UPDATE! $4.95 April 1990

ComputerSystems

Take Out to the Limit

A Quarterly Magazine for the users

EXPLORING THE POWER

OF

SINCLAIR COMPUTERS

THE CAMBRIDGE Z88
Portable Power

THE SINCLAIR QL
32 Bit Power House

THE SINCLAIR TIMEX TS-2068
Versatile Desk Topper

In this issue. . .

UPDATE CONTINUES WITH NEW PUBLISHER

QL OPERATING MS-DOS *TS-2068 EXTRA MEMORY* *USING RAM DISK with TS-2068*

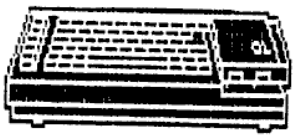
PSION'S THREE NEW MOBIL COMPUTERS *NEW Z88 PUBLICATION* *QL'S TEXT '87*

PLUS- NEW ISSUE DISKS FOR BOTH THE SINCLAIR QL AND THE TS-2068

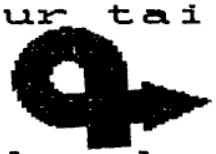
UPDATE MAGAZINE
1317 Stratford Ave.
Panama City, FL 32404
904 871 3556

NEWS AND NEW PRODUCTS

If you like your Sinclair



Don't chase your tail···

Turn it around  and get your tail over to the Milwaukee Sinclair Expo

The 1990 SINCLAIR COMPUTER Exposition

MILWAUKEE WISCONSIN

June 2nd & 3rd

Banquet Friday Night June 1st

- SEMINARS, SOFTWARE, HARDWARE, DOOR PRIZES, SWAP SHOP -

Location - NAUKESHA HOLIDAY INN - Hwy 18 & I94 Naukesha WI

Saturday June 2, 1990 will be the SNUG meeting

Tickets: in adv./at door
One Day - 4.00 5.00
Two days- 7.00 8.00
Banquet -16.00 ????
Tables: 30" x 6' 25.00 each

If flying in call us, 414 353 4522,
with flight info. for transportation.
Mail your reservations to:
Expo Reservations, P.O. Box 101,Butler WI 53225

UPDATE CHANGES PUBLISHER

by Bill Jones

This announces that Frank Davis, 513 East Main Street, Peru, IN 46970 will be taking over as publisher of Update Magazine with the October 1990 issue. We hope that the subscribers will not notice any change in quality or depth of coverage except for the better. I am delighted that Update will be a survivor publication and will be published under such competent leadership as Frank Davis can provide. I will help Frank as much as he requests to keep Update an excellent publication.

Frank Davis is a long time user of both the TS-2068 and Sinclair QL. He has recently been involved as the SNUG Librarian for public domain software. Frank was a key person in organization of the 1988 Computer Fest at Indianapolis and has been the editor of a TSUG newsletter. In addition Frank is in close coordination with other SNUG officials in Indiana who will be at hand to assist in keeping Update Magazine a good publication.

For those of you who have articles to publish please keep on sending them in. I will turn over all un-published articles to Frank when the July issue is finished. I am looking forward to having more time to spend on programming and my sailing hobby. Also I promised to take the Wife on a cruise in the Virgin Isles.

The hints given in Update recently that I was about to quit publishing brought on many letters urging me on. That was enough to make me reach out to find someone to carry on the magazine. But how can one sell ice to Eskimoes, Confederate currency, an old treasure map, a sunken boat, or a non-profit magazine? But the effort paid off. Now we have another dedicated publisher to keep Update going, the first Sinclair Timex magazine to survive the quits. I urge all to give Frank Davis your support. Re-subscribe and try to bring in another Sinclair Timex user as a subscriber. There are many inactive S-T "owners who are not users". We need to find these folks and encouraged them to subscribe to Update and re-join this enthusiastic group. The subscribers really are an enthusiastic group having a convival spirit of friendship with a common interest.

Update is your magazine. There will be some complexion changes, likely for the better. Each publisher injects his own personality into the pages. Since all Sinclar - Timex supporters are eccentric characters beyond help from couch analyists, Frank Davis will likely bring in a new set of egocentrics and incomprehensibles for you to enjoy. I quit with the hope that you are still trying to figure out whether or not I have any literary talents or am truly the curmudgeon that one of our subscribers suggested. My intent all along was to confuse while injecting humor and a bit of satire to twit the perfectionists who look for nits in the forest of verbs, pornouns, and infinitudes, and find many because Update is too poor to hire a proof reader. Its the beef that counts.

I also would like to remind everyone that Update Magazine is not a profit venture. The best that a publisher can hope to do is to avoid losing money. The best that I could do during the three years was to put aside enough funds to pay for the upkeep and entertainment on board the 200' yacht "White Star" as we plyed the seas between issues. (Now I'll tell you. The "White Star" is a 45' sailing yawl, built in 1936 at Bristol, Conn. It was our family vessel for 10 years and was given to a boy's club in 1968 when I was sent to serve my Country in Viet Nam. Sailing gave way to publishing three years ago and will be resumed with salty zest.)

A good gesture of support to your new publisher would be to send him your subscription renewal for 1991, which becomes due after the July issue is received. I wish Frank Davis and the subscribers to Update good tidings. -BJ-

UPDATE Gets a New Editor After July

In the January issue of UPDATE, it was announced that if no one could be found to carry on the publishing and editing of UPDATE magazine that it would be discontinued with the July issue (the last issue on the subscription year). Bill Jones was getting a bit burned out and wanted to spend more time at his other hobby, boating. It has been an excellent magazine and has a fairly steady subscriber base. The magazine had been twice offered to SNUG, but was not accepted at that time. At this time SNUG plans to continue with its own newsletter, of which the second issue is probably already sent out.

Well the gauntlet was thrown down and it has been picked up, by Carol and Frank Davis of Peru, Indiana. After the July issue is sent out from Panama City, Florida, the magazine will be moved to Peru, Indiana. This means that all renewals of subscriptions will go to UPDATE Computer Systems, P.O. Box 1095, Peru, IN 46970. They will also, after the July issue is mailed, be handling all orders for program disks from the magazine. In the past this magazine has had near 100% renewal, and it is hoped this will be continued. Most of the magazine will remain pretty much the same, other than adding a column on Sinclair desktop publishing. We hope to retain all of the same writers and perhaps add a few. At this time all prices and policies remain the same. The price for a subscription is $18.00 U.S. The only change being looked at is that , with a postage rate hike likely in February 1991, we may have to increase postage for those subscribing overseas and in Canada by a dollar or two. It costs more to mail out of the country. All correspondence or questions on this should be directed to Frank Davis at P.O. Box 1095, Peru, IN 46970. Assisting the Davis family with this venture is their longtime friend, Eliad P. Wannum, who will be active in the editorial department. He will undoubtedly also get stuck with a lot of collating and stapling of the magazine.

For those who do not know Frank Davis and his wife Carol, allow me to say a few words. Remember the first Midwest Timex-Sinclair Computer Fest at Cincinnati, Ohio? He was the guy who got it all started and together with Jack Roberts and a committee of 11 more, put on that show. The next year he went on to produce the 2nd Midwest Timex-Sinclair Computer Fest in Indianapolis, Indiana. To this date, that was the largest and best attended Sinclair show on this side of the Atlantic. Those who attended the show the next year at Cleveland will have remembered he was one of the moderators for the large meeting for all attendees and that he presented a seminar on data exchange between the TS1000, TS2068 and the QL. For the last four years he has been the Editor of the ISTUG Newsletter, except for a few months that Mike Felerski had the helm. He was one of the original founders of the Indiana Sinclair-Timex Users Group, in which he is still very active. Presently, he is the Head Librarian for SNUG. In each and all of these endeavors, he has had the support of his wife, Carol, serving on all committees, proofreading, typing , etc.

We will look forward to having the full support of all subscribers, writers, programmers and Sinclair vendors. Thanks to Bill Jones for giving us another chance to serve the Timex and Sinclair users of North America.

PSION DEVELOPS THREE NEW COMPUTERS

A new generation of MOBIL full capability Computers!

(To prounounce PSION, drop the "P" and say, 'si-aan').

PSION has been a leader in Software development since the first Personal computers were introduced by Sinclair. Some of PSION'S classic softwares are: View Calc and View File for the TS-1000, TS-2068 and Spectrum, Quill, Abacus, Archive, and Easel, for the Sinclair QL. PSION is an American Company based in Watertown, Connecticut, and they are producing a new line of computers.

First came a PSION "Minature", a calculator size computer having full computing capability and the ability to transport files to and from IBM compatables.

NOW A NEW SERIES OF "MOBIL" COMPUTERS

Now PSION comes in with a new line of "MOBIL" Computers that have many new and attractive innovations. PSION coined the name "MOBIL COMPUTER" to better discribe the size and portability of this new group of computers. The terms "Portable" and "Personal" had become meaningless applying to a wide range of computer sizes. Three new computers are designed to have FULL DESK TOP CAPABILITY.

PSION'S "Mobil" computers are the size of a loose leaf note book, measuring two inches thick and weighs 4 1/2 pounds. The three models have clam shell design to provide a full size screen that folds up to provide high resolution viewing with built-in graphics interface.

The three models of PSION Mobils are designated: MC-200, MC-400, and MC-600. The first two are designed with PSION'S own Operating System. The MC-600 is designed to use MS-DOS and all of the software used by the IBM compatable market.

Common Design Features

All three of the PSION Mobils employ the newest technology of "Flash Memory Solid State Disk" (SSD). These SSD's are match book size Flash Memory cells that act in the same manner as disk drives, but have no moving parts. Each of the PSION Mobils have four SSD's built in.

The PSION Operating System is ROM BASED, meaning that no outside DOS utilities are required for operation.

ALL RAM MEMORY is held Non-Volatile by replacable AA batteries that last for up to 75 computer operating hours. In terms of normal usage this translates into months of operation.

THE MC-200 - MC-400 Mobil Computers

These computers use the PSION Operating System and have an ensemble of BUILT-IN Softwares including, Word Processing, Personal and Business Data Base Management, Personal Diary, Full Function Calculator, and Terminal Software for Telecommunication.

The MC-200 and MC-400 have SPEECH PROCESSING. A built-in Microphone is used to input speech which is stored in memory. Speech processing includes its editing. Entrys to the built in software can be made by voice. Play back is also provided.

The MC-200 and MC-400 have a built-in "Touch Pad", a Mouse atop the keyboard. The Touch Pad is integrated with the built-in softwares to provide menu item high lighting for selections.

The MC-200 and MC-400 use the PSION "FAST LINK" to transport files to and from either IBM or Apple computers. Fast Link will be discussed more as the MC-600 Mobile is described.

THE MC-600 MS-DOS MOBIL COMPUTER

The MC-600 is designed for full compatability with the widely used IBM compatable systems. The MS-DOS 3.2 DOS is built-in to the MC-600 ROM. PSION is currently adapting its own Library of software into the SSD format. Transportability of MS-DOS disk files and programs is provided by a built-in PSION Communications Link. The built-in software links any MS-DOS computer via PSION's own "Fast Link", a serial interface that transfers data at a rate of 1.5 megabytes per second. Fast Link works with both IBM and Apple Computers.

Remember, PSION has been a full capability Software Enterprise since the beginning of the Microprocessor Computer industry. Their vast experience and library of software provide a combination that is unique in the Computer Design industry. Most Computer industries build their computers and depend upon outside software support of their systems. That PSION is converting their library of software for this new line of Mobils promises un-excelled software support of these computers for years to come.

While the PSION MC-600 uses MS-DOS which has wide software support, most new computers that have new Operating systems suffer from lack of software support. PSION's in-house software capability promises full software support of the MC-200 and MC-400 from their beginning. (See EMSOFT Ad for specs.) -BJ_

To a hundred who wrote: *The January issue of Update was mailed on March 4. Not anything to be proud of for sure. Excuse: My apple cart began to tilt just after Christmas, first with the need for removal of a colon polllup, then an angioplasty operation to restore a clogged artery, and then a kidney procedure to blast away a boulder. All this took 32 days in ye ole hospital. I began to think that I'd have to put in a change of address card to the postal service. In the meantime about 100 letters were received asking "Wheres my magazine?" (Wheres the beef?). A dozen or so had $18 checks inside, the writer thinking that perhaps he had forgotten to renew. These checks were returned as all writers were current subscribers. I'm back in the pink again now, so dont send flowers. But the bills are beginning to arrive which may cause worse pain than did the kidney stone. I thank everyone for the pleasantries inclosed in the letters. (No one sent onions.)*

HAROLD A.ROMER
P.O.BOX 32
JOHNSON CITY,TENN.
37605-0032

I have for sale the following items:

2 - ZX80	-with power supply/manual	$25.00 ea
2 - TS1000	-with power supply/manual	45.00 ea
1 - ZX81	-with power supply/manual	35.00
1 - ZX81 With klik keyboard-w/ps & manual		40.00
2 - 16K memory(1016)		15.00 ea
1 - 32K memory (Memotech)		30.00
1 - 64K Memory (Memotech)		40.00
1 - Hunter board W/4 chips		35.00
1 - Eprom pgmr(MAV) for ZX81/TS1000	2/2732	25.00
1 - 40 col ptr		40.00
1 - 32 col ptr (2040)		40.00
1 - Mannesmann Tally Spirit 80 Ptr.		150.00
1 - Memotech keybd/81-1000		55.00
1 - demo hardware/1000		15.00
1 - Carry case/81/1000		10.00
1 - OS64 Cart-64col		15.00
1 - Modem for 81/1000/2068		75.00
1 - AERCO ptr cable W/Tape		30.00
1 - Memotech ptr cable 81/1000		45.00

##########

IT'S HERE!!

QL TRIVIA-QLUE

HERE IS GREAT NEW COMBINATION GAME!
IF YOU LIKED QLUE AND TRIVIA, THEN THIS IS FOR YOU!
WE HAVE COMBINED THESE TWO GREAT THINKING MANS' GAMES
IN ONE GREAT PACKAGE!!

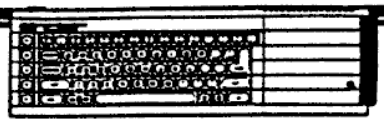
13 ROOMS - 13 WEAPONS - 13 SUSPECTS
THAT'S OVER 2100 POSSIBLE OUTCOMES!
AND NOW - OVER 750 TRIVIA QUESTIONS!
QUESTIONS COVER EVERY CONCEIVABLE SUBJECT!
RELIGION - HISTORY - SCIENCES - GEOGRAPHY - ASTRONOMY
YOU'VE NEVER SEEN QUESTIONS LIKE THESE!

Also included, at no extra cost, your choice of
FOOTBALL FORECASTER or LIFE EXPECTANCY FORECASTER
(Specify which you would like when you order.)

WE THINK YOU'LL LIKE THIS ONE!
OUR PRICE? ONLY **$19.95** +42ph

AVAILABLE EXCLUSIVELY FROM:
RMG ENTERPRISES

##########



We carry **ALL** hardware and software lines for the Sinclair QL.

We also carry the Z88.

WRITE FOR FREE CATALOG

Sharp's, Inc.

Rt. 10, Box 459
Mechanicsville, VA 23111
(804) 746-1664 or 730-9697

NEW!

Compare the PSION MOBIL COMPUTER to any other portable computer now in production.

Dollar for Dollar, feature for feature, it can't be beat.

Model	MC-200	MC-400	MC-600
CPU	**** 80C86 16 bit CMOS ****		
Speed (Mhz)	7.68	7.68	7.68
BIOS mfg	Phoenix	Phoenix	Phoenix
RAM Standard	128K	256K	768K
RAM Maximum	512K	512K	768K
Screen diag	10"	10"	10"
type (LCD)	SupTwist	Retardation	Film
protocol	CGA	DCGA	DCGA
resolution	640x200	640x400	640x400
col x line	85 x 14	85 x 14	???
No. of Keys	69	69	79
Detach keybd	no	no	no
Slots Stand	0	0	0
Prop	2	2	2
Battery type	8xAA Alk/Opt Rech NiCad Pak		
life	75 hr	60 hr	40 hr
replace	yes	yes	yes
Ports Ser/Par	1/1	1/1	1/1
Video Port	no	no	yes
ROM size	256K	256K	256K
Software in ROM	GUI	GUI	MS-DOS 3.2
Touch pad	yes	yes	no
Floppy Stand	no	no	no
Floppy Ext.	no	no	1.44Mb 3.5"
Hard Disk	no	no	no
RAM Disk	OK	OK	1 Mb intern
Maximum	2 Mb	2 Mb	8 Mb
Slots	4	4	4
Fast serial transfer	***** External Option *****		
Modem	***** External Option *****		
Microphone	yes	yes	no
Speaker	yes	yes	yes
Programing language	OPL	OPL	n/a
Dimension	*** 12.4" x 8.9" x 1.9" ***		
Weight (lbs)	4.3	4.3	4.3
Warranty	******** 12 months ********		
Price	$1199	$1699	$2999

Call for current quote from
EMSoft
Box 8763, Boston, MA 02114
(617) 889-0830

A New Z88 publication has surfaced. Z88 Fax News is propagated by Fax line or mailed to subscribers who do not have FAX. One of the four pages of the issue received is duplicated below. Update Magazine presents this as NEWS ONLY and does not necessarily endorse the Z88 or Z88 FAX NEWS.

Z88 FAX NEWS

Publisher et al.....Curtis Claar
Facsimile.....Promo Productions
2342 Glendale Ave. • Casper, Wyo. 82601
Fax or voice (307) 766-7026

February 1990 Page Three

Cambridge N. America

Is cash a problem?

So is Cambridge North America broke or aren't they?

One CNA employee was quoted as saying, "We are very alive." Others agree.

But reports still keep coming in of financial dire straits. Some say CNA owes Cambridge UK more than $1 million. Others say CNA has bought more ads than it can afford. Some people say CNA is simply bankrupt.

As is often true in researching the records of a company, researching CNA has proved to be no less than confusing.

The search for the truth begins with CNA's name.

Marc Giguere of CNA said CNA is merely an assumed name for SSI Computer Systems, Inc., a Delaware company.

SSI is not to be confused with SCI. SCI is the company that makes the Z88 and IBM and Wang keyboards, he said.

Giguere said Diversified Foods owned New England Sales, Unicorn Transportation, and Cambridge North America.

He said that 4-5 months ago, Diversified Foods was bought by the Bank of Boston and Cambridge North America subsequently separated from the conglomerate. He said

see CNA page three

Z88s not just for little guys

It's small, but 'big' people still use it

The Z88 has always been considered by non-Z88ers to be powerless just because of its size. But several famous people have proven size isn't everything. Many people that are in the public eye are buying laptop computers and the choice for some has been the Z88.

Singer Mel Torme, the jazz duo Tuck & Patti, comedian/magician Teller of Penn and Teller, and Anchorage Daily News Editor Howard Weaver are all in the ranks of those who own Z88s.

Mel Torme uses a Z88 and it's said that he is currently writing a book on it — that is using it, not about it. Cambridge North America has also reportedly approached him about a making a televsion commercial for them. It is still unknown what will come of this.

Tuck of Tuck & Patti, an internationally famous jazz duo, is said to be a real power user. Tuck has a 512K RAM card in the first two ports and several EPROMs to use in the third. Tuck supposedly uses it all the time and takes it everywhere.

Teller, the silent half of Penn and Teller from New York City, surfaced on the Z88 BBS in Torrance, Calif. (BBS phone number: (213) 370-2754). Teller has a Z88, but not much else it certain — after all, he is the quiet one. Penn and Teller, though, apparently have been computer buffs for some time. At one time they — or some real good fans —, were running a Penn and Teller Fan Club BBS.

And even Howard Weaver, the editor of the Anchorage Daily News, uses a Z88. He has written a few interesting programs for it and is considering buying Z88s for his reporters.

Z88 Fax News will keep its readers updated on famous Z88 owners. Watch for more articles on this subject in the next few months.

— Curtis Claar

Welsh company sells linkup packages

The Music Suite Ltd. in Wales is now offering two linkup packages for the Z88.

They have been offering a hardware and software linkup package for the Atari ST for nearly two years now and last year they came out with one for the Amiga. They have plans for a PC version as well.

The Music Suite is run by Adrian and Helen Wagner and Francis Monkman.

"It has always been our intention to fully support our software," Adrian said in a fax letter last month. Because of this, they claim to offer more than simple dump programs.

According to advertising, the programs also route Z88 printer output to printers connected to the ST or the Amiga.

A review is forthcoming.

The Z88 and Telecommunications

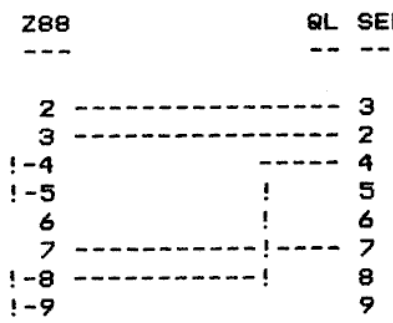
by Dave Bennett

The Z88 is particularly well suited for communicating to modems and other computers. A serial port which communicates up to 38,400 baud, a VT52 terminal program and a dedicated file transfer utility are included with the stock Z88. The Z88 also weighs only 2 pounds and is easily transported right next to the device you want to communicate with.

My Z88 is mainly used to talk to a pocket modem and a QL. Z88 files are archived on a QL disk. I also occasionally communicate with PCs at work and other Z88s.

I have designed and built cables to communicate with the QL, PCs and other Z88s. I built the modem cable from recommended pinouts by Cambridge.

The QL cable is as follows:



I have tried several other designs for a Z88 - QL cable but have not found a cable that works as well for transfering files.

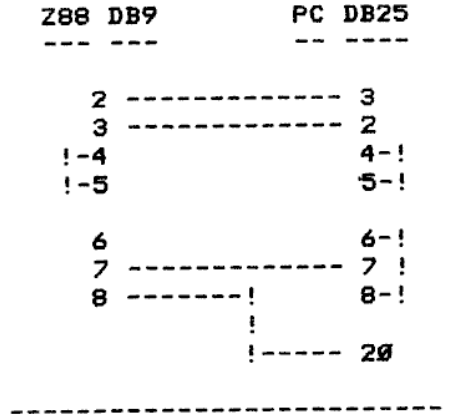
What I use to transfer files are Q_Link for the QL and Z88COM for the Z88. Q_Link is a shareware terminal program from Harvey Taylor. It includes CRC and Checksum xmodem as well as many other features. Z88COM is a publicly available terminal program from Phil Wheeler. It extends the built-in VT52 with checksum xmodem and some other features. Phil also operates the Z88 BBS in Los Angeles.

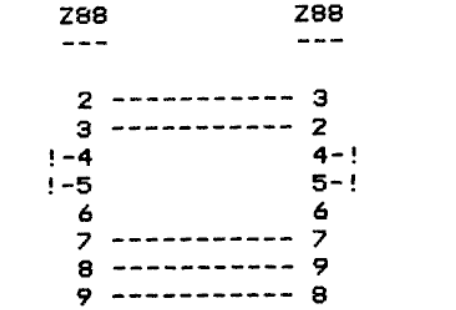
I basically connect the computers with the cable and run terminal programs which utilize xmodem on each machine. You set up one machine to send and one to receive. Then you start the transfer on both machines at the same time.

You can transfer ascii files without xmodem by various means but I prefer to use xmodem. It provides error checking in the transfer so you know the files are transfered OK. You can also transfer any type of file even machine language.

Z88 to PC and Z88 to Z88 transfers work similarly. On the PC I prefer to use PROCOMM. It is a powerful program and extremely easy to use. I would like to see it made available for the QL. For Z88 to Z88 transfers I am using Z88COM on both machines.

The Z88 - PC and Z88 - Z88 cables are as follows:





There also is a variation on Z88COM called Link by Dave Armstrong which turns the Z88 into a mini BBS. There are no message facilities but you can transfer files. All typing is done at a remote computer. The Z88 just sits

there. It is similar to PROCOMMs Host Mode. It works well connected to a computer in the same room. Modem facilities were removed from this variation but you can connect to the modem and then switch to Link by hand. With this program you can transfer files with someone across town at a prearranged time. You could even transfer QL and 2068 files.

There also is another program available for the Z88 called ZFU by Richard Haw. This is a utility which takes many files and archives them into one large file. In the process it compresses the files so the resulting large file is much smaller. You can archive all the files in all the ram devices of the Z88 into one file and just backup that one file to your QL or whatever.

The Z88 also works nicely with any Hayes compatible modem.

Here are the official modem pinouts from Cambridge:

Z88		Modem
2	----------	2
3	----------	3
7	----------	7
5	----------	20
8	----------	20
9	----------	20

The Z88 includes a built-in VT52 terminal emulator. It works quite nicely although you have to dial manually with Hayes AT commands. It is also easy to send and receive ascii files with just the built-in software. The built-in VT52 also connects quite nicely to a VAX mini computer once you tell the VAX you have a VT52 with 8 lines.

But I like Xmodem. A terminal program was developed in England which provides xmodem. The disadvantages of this program are its high cost and it also adds its own protocol in place of the VT52. The built-in VT52 works much better.

Fortunately Z88COM came out which provides xmodem but works in concert with the built-in VT52. Z88COM is also much cheaper. In most cases it is available for the cost of a download. Z88COM can also autodial and autologon to five systems. It has a message upload feature for sending text files. All in all it is a great program.

The latest I heard was that Cambridge just came out with a VT100 terminal program. I have not heard too much about this. The VT100 can display graphics from a remote IBM or VAX. It moves the cursor around your screen on command from the remote system. I am not too sure how well this will work on the Z88 with its 8 line screen but it sounds interesting.

The Z88s portability and sheer connectivity to anything speaking RS232 make it a great little computer for telecommunications.

HARK Z88 USERS: (WHAT A NICE GUY!!)

Dear Bill, I have been trying to promote the Z88 for about two years. I have about 350K of Z88 programs and related QL programs that were downloaded from two main Z88 BBS: 703-968-9438, 213-370-2754, and GEnie.

I can put the files on a QL 720 sector disk or a IBM 360K disk. In the case of the IBM disk I will substitute some IBM programs instead of QL programs. See below:

Z88 Public Domain and Shareware Programs:

(Approx 350K of files)

Available on QL DSDD 720 sector disk or IBM 360K disk. The QL disk includes Z88 related QL programs. The IBM disk includes IBM related files.

PRICE: A dollar to pay for mailing, or two dollars to mail in a disk mailer.

David Bennett
329 R. Walton St.
Lemoyne, PA 17043

THE SINCLAIR QL

(BITCH! BITCH! AND DO NOTHING)

"My Sinclair QL is so un-reliable that I cannot get a simple one page letter done without the QL crashing. The crash usually happens when the refrigerator or furnace kicks on." This complaint was received a year ago from a QL Subscriber. Still, almost every week I talk to a QL user who is still having such problems in varying degree. THERE IS NO EXCUSE FOR HAVING SUCH PROBLEMS ANY MORE!! Read on--

March 17, 1990

Bill Jones,

The Q/L on which this letter is being composed has been modified per Dan Elliott's article. (July 1989) The lock up problems associated with the Q/L power supply plaque me no more. There is no longer a need to save text every few minutes. The only lock up experienced now is caused by conflicts between the software and the inputter.

The Coleco power supply is enclosed in a fabricated, black case, with openings for air flow and is now sitting on the floor providing cool uninterrupted power. This modification is not difficult, it's not expensive, and is highly recommended. Tell everyone to do it themselves or call Dan, but get it done!

My confidence in the Q/L is now such that I am in the process of putting together a disk drive system and saving for a Trump Card expansion. The microdrives work fine, usually, but I would prefer a faster, more reliable system.

Ha !! The furnace just came on and someone started the microwave and I am still input'in. To bring this to a close, I thank you and all who contribute to the cause of keeping the T/S computers working; the writers, programmers, and dealers.

Sincerly,

DRJ

Donald R. Jones
1900 Sunvale
Olathe, KS 66062

Dan Elliott
RR-1, Box 117
Cabool, MO 65689
Tel 417 469-4571
(weekends)

Repairs all Computers
Specializes in repair
of QL and TS-2068
Power mods for both

(MY Coleco PS SITS under The MONITOR.)

Ed note: Need I say more? But I will. Some QL users get the downs because they are affected by the dis-respect from the Yuppies, the ridicule from the $5000 IBM users, the syndrome "I only paid $100 for it so it must not be very good", and the "Sinclair abandoned us syndrome". ANYTHING to bolster insecurity and lack of self esteem. But others are bragging that their QL does 10 times as much as the IBM AT. I guess that is the difference between the pessimist and the optimist. You know what I mean. The pessimist believes that the sky is falling upon him, while the optimist believes that the sky is rising with him in it. Me? I believe that when one has a problem he should fix it. When it is forecast to rain I leave my hat behind as I go fishing. On Sunday I believe that the preacher will soon quit talking. I'm always sure next months bills will be less, that Ed McMann will send me that million that he promises, and the Wife wont have a headache. Don, my QL is working fine too! But work on that interface conflict, and you'll overcome. -BJ_

Ruminations From The QL Editor

There is a lot of interest among QL users about the PC world and MSDOS operating system. Its makes sense because the influence of PCs on personal computing is tremendous. Digital Precisions emulator, PC-CONQUEROR, makes it possible for any QL user with disk drives and memory expansion to turn their QL into a quasi PC. This is certainly a novel capability and has useful applications but be aware that the QL makes for a SLOW PC. The QL is a fine computer. With disk drives, memory expansion, or TRUMPCARD it can still hold its own against a PC. It is slower in many ways but has capabilities that most PCs do not. If you use your QL effectively you can still work as fast or faster on it than most PC users can on their computers. It can serve you well for years to come. Learning to use your QL effectively probably makes more sense than playing around with MSDOS on it. The purpose of UPDATE Magazine is to help you reach that maximum potential with your QL.

That said, I need to explain why we are giving so much coverage to MSDOS in this issue. It's definitely not because we feel that the QL is dead and we all need to swtich to MSDOS. It's because we want to provide information to our readers. There are many QL users who are curious about MSDOS and many others who work in both the QL and PC worlds. UPDATE needs to check out QL Software related to PCs such as : DISCOVER (allows QL PC disk compatility, $40), PC CONQUEROR (PC emulator and allows QL PC disk compatibility, $150??). PC software related to the QL include : PSION's PC-FOUR (PC versions of QUILL, ABACUS, EASEL, and ARCHIVE,$99).

This issue is packed. Part III of David McCall's WINDOW DEFINING series focuses on SUPERBASIC procedures.

Peter Hale starts a 2 part series on TEXT87. Al feng clears the air about ROMs on the QL.

Bob Hartung shows how to set up an ARCHIVE address database.

Bill Jones shows us some QUILL printer driver tricks.

We have lots of coverage of PC CONQUEOR and MSDOS with articles by both Bob Hartung and Al Feng.

There are a couple of articles about the TS-2068 that the QL users may benefit from. One is about Dot Matrix printer drivers. The other is a Business Software for the TS-2068, done in Basic. This excellent program offers a challange to the QL users to clone it in Super Basic. The benefits will be worth the effort and the conversion should be good practice.

Finally, I explore the QL-PC connection through ARCHIVE by showing how I converted one of my ARCHIVE programs to work on a PC. And there is more.

Next issue David McCall will tie up his Window Defining series and we will have an Issue Disk of it. Peter Hale will continue his in depth description of TEXT87. More of Timothy Stoddard's TRUMP CARD Manual.

Don't forget about the 1990 Sinclair Computer Exposition in Milwaukee hosted by SMUG on the weekend on June 2&3. See You there.

WOOD AND WIND COMPUTING

Bill Cable : RR 3 Box 92 : Cornish, NH 03745

CABLE COLUMN

QL - ARCHIVE - PC

Databases are specialized programs for efficient handling of both text and numeric information on a computer. Anyone trying to put their computer to good use should be using a database in one form or another. As many of you know I am a great fan of the ARCHIVE database and programming language. I write commercial software with it. For handling information on a QL it has no peer for its capabilities. This is not to say that there is not a price to pay for using ARCHIVE. Either you have to learn how to use its programming language or you have to buy ARCHIVE programs. Other alternatives include buying FLASHBACK or write or buy database type programs written in SUPERBASIC or other languages.

In earlier issues of UPDATE I have tried to explain the basics of using ARCHIVE. Now that I see there is so much interest in MSDOS by QL users I want to push ARCHIVE from another angle. If you want to be able to do something useful on MSDOS computers then learn the ARCHIVE language on your QL. I have been aware that there is PC version of ARCHIVE. But until recently I have had very little experience on MSDOS machines (PCs). After tackling my first serious conversion attempt (making DBEasy work on a PC) I would say I am already close to being an expert with PC-ARCHIVE because the PC and QL versions are so similar. Almost all of the ARCHIVE I have learned on the QL can be transfered directly to the PC. DBEasy, my ARCHIVE database manager, is now available in both QL and PC versions (see my ad). The importance of this QL-PC connection should not be minimized. Transporting software and experience from one computer to another is generally a very difficult process. If you take an elaborate SUPERBASIC program and try to convert it to run on a PC using one of the PC BASIC languages, almost certainly you would have to make major changes to the code and a lot of learning to do as well. So learn the ARCHIVE language on your QL and you can apply what you have learned directly to a PC. You can transfer your databases and programs to a PC as well. And its not like ARCHIVE is an inferior language. I still maintain that it is one of the nicest languages around. Its a nice structured procedure oriented language with a built in editor and good error messages and handling. Its very much like SUPERBASIC without line numbers or bugs. And I have talked with people who are experienced with PC-ARCHIVE and other PC databases and they feel that ARCHIVE is a superior database for PCs. Just like the QL was/is ignored by the masses, PC-ARCHIVE is ignored even though it is a fine database. And compared to the price of DBASE or PC database programs it is a bargain.

This Cable Column is going to describe how I went about converting my DBEasy program to the PC. This included purchasing a PC computer, the Amstrad PPC640 portable, and learning a little bit about using MSDOS. I will also give some statistics comparing DBEasy's performance on the QL and on different PCs. We already have ARCHIVE for our QLs. For the PC you must purchase it as a package with QUILL, ABACUS, and EASEL called PC-FOUR. The price is about $99. It can be purchased from me (see my ad) or EMSOFT or RMG or Sharps. If you work on a PC it makes sense for you to have PC-FOUR because you

use the same programs on your QL. The PC versions also have extra features. It's pure folly to purchase PC-FOUR to use soley with PC-CONQUEOR. The QL makes a poor PC.

Choosing a PC

Before I could do serious PC work I needed a PC computer. DBEasy swamps PC-CONQUEROR so I couldn't do the work on my QL with the emulator. Even if it worked, life is too short for its snails pace. The possible choices for a PC are limitless. They range from an XT type (8088 or 8086 processor) or an AT type (80286) or an 80386 machine with prices from $500 to $2000. My final choice was a portable by Amstrad, the PPC640. It is an XT type but the 8086 processor has a 16 bit data bus so it is almost as fast as an AT of similar clock speed (8 MHz). It is too heavy to be a laptop but is totally portable and takes little space to set up. It will run from batteries or 12 volt DC (as in a car) or normal house current. I have my own home power system and the fact that it uses only 20 watts of power and I can run it on 12 volt DC were big pluses to me. I also liked it having a full size AT style keyboard. It has two 720k 3 1/2 inch drives. It has an LCD screen (no backlight) but will also plug into my Magnavox RGB monitor with CGA display. I use the same monitor with my QL. The PPC640 also has a built in 1200 baud modem with phone connector and is provided with software to run it. It has 2 serial ports and 1 parallel port. I bought it from UNDER-WARE Electronics (800-442-1408) as advertised in Computer Shopper for $700 plus $25 shipping. Although still available, it is no longer in production and was originally suppose to sell for something like $1200. There are also versions with one disk drive and no modem available for less. It comes with DOS 3.2. It has worked flawlessly for me and I recommend it. The keyboard action is not the best but works fine. The drives are quiet. The LCD screen needs good room lighting but is adequate. I have found it to be fast compared to my QL mostly because of much faster screen handling. There is suppose to be a 20 Meg portable hard drive available for $300 but I don't have it yet. It is certainly an inexpensive way to break into the PC market place. I have a comprehensive payables/receivables package for home and small businesses written in QL-ARCHIVE that is almost ready for release. By converting it to PC-ARCHIVE, I have a package to sell locally to small businesses. Later if things go well I will consider getting a more versatile 386 machine. Have I abandoned my QL? No way! I consider QDOS a more flexible operating system than MSDOS and I like being able to multitask. I don't want to abandon SUPERBASIC with its excellent language for making quick programs with very flexible graphics. Even if slower in some respects, I can still work faster on my QL because of its excellent features. In the future I will work on both the QL and PC machines. It is certainly nice to have the 4 PSION programs available on both.

The DBEasy Program

DBEasy is a front end database manager for ARCHIVE. It allows you to use most of ARCHIVE's features without having to know anything about its language. It allows the user to create and manage all sorts of databases. It is a large complex program (55k on the QL and 56k on the PC) but very simple to use. As with any friendly software most of the development effort went into making clear and simple menus and displays.

Transfering DBEasy to a PC

Now that I had a QL, a PC, ARCHIVE for both I next needed to transfer my QL files to the PC. There are at least 2 ways available. The best way is to use a QL software like DISCOVER or PC-CONQUEROR to copy them to a PC formated disk on the QL and take the disk to my PC. I couldn't do that because my QL has 5 1/4 drives and my PC has 3 1/2 drives. I have 3 1/2 drives on order for my QL from RMG to make life easier. So I had to use the serial ports on my QL and PC to transfer the files. I used the communication software, MIRROR II, on my Amstrad to communicate with my QL which had software so it emulated a VT-52 terminal. I had some trouble figuring out the cabling. Peter Hale provided the necessary expertise to finally get it working. After that it was simple. Only text (ASCII) files can be transferred. This means I have to have my ARCHIVE programs in _prg form rather than _pro form. If you save a program in ARCHIVE with the SAVE "filename" command then it is saved in ASCII format (_prg). If you save it with the SAVE OBJECT "filename" command then it is saved in binary format(_pro). Normally you keep large programs in binary format because they load faster but for transfering to the PC I had to put them all in ASCII files. Also my databases had to be in export (_exp) files (ASCII) rather than database (_dbf) files for the same reason. Since screens (_scn) are not ASCII files I can't transfer then. I must reconstruct them with the ARCHIVE SEDIT command on my PC. After transferring I started PC-ARCHIVE, imported the databases, rebuilt my screens, started DBEasy and changed code where problems existed.

Some differences between QL and PC ARCHIVE :

The most important difference for me is that there is no POSITION command and RECNUM() function in PC-ARCHIVE. I like to use those features to keep track of where I am in a database. If you use the FIND command it is nice to know how far you are in the file if you find something. They have been replaced in PC-ARCHIVE by a more powerful ordering and selecting process that allows you to keep 4 ordering and 1 selecting definitions at the same time and switch between them. This is a very powerful feature but my code doesn't take advantage of it yet. The BACKUP command works differently in the 2 versions. On the QL it is similar to a general file copy command but on the PC it only works on databases and does a record by record copy. In PC-ARCHIVE there is no FORMAT command. There is no RENAME command in QL-ARCHIVE. The PC-ARCHIVE has more complexities with paper and ink colors because mono and color monitors display differently. PC-ARCHIVE and the other PSION programs have a special task sequencing language (TSL) which can be used to control the program at startup. This allows me to have ARCHIVE and DBEasy start up automatically, something I can't do with QL-ARCHIVE. These are the major differences I have discovered when I tried to run DBEasy on the PC. The hardest part was dealing with all occurences of RECNUM() and POSITION in my code. I have worked out a display format on the QL which works surprising well on the PC so the DBEasy program looks exactly the same on the PC. Now that I know more about the differences I can plan ahead so conversions will be much easier. Since PC-ARCHIVE has more features than QL-ARCHIVE it will not be as easy to transfer back to the QL once I take advantage of those features.

DBEasy Performance Chart (time in seconds)

DBEasy Task	QL 7Mhz	IBM PC 4Mhz	PPC640 8Mhz	AT 10 MHz	AT 20MHz
	- Computer Type -				
Load	69	94	57	48	10
Find "xxx"	15	17	11	9	2
Step 10 records	58	50	19	12	8
Display new page	17	20	7	5	2
Sum Population	13	9	4	2	1
Order by 'country'	13	18	9	7	1
Locate 'SWEDEN'	4	4	2	9	2
Select 'Europe'	19	24	12	7	2

In the chart above I give elapsed time for DBEasy to do various tasks on a few different computers. I used my DBEasy program with the gazet database (155 countries of the world) modified for DBEasy. The time is in seconds. The 20 MHz AT used DBEasy from hard drive. On all the others it ran from disk drive. I must say that using DBEasy on the 20MHz machine was almost intimidating. You couldn't slow it down by doing anything. Some may be horrified to see how slow the QL is. But except for the tasks which require heavy screen work the QL gives very acceptable responses. And those that were slow like stepping through 10 records are adequate because you wouldn't be stepping through the records unless you wanted to read them and you need that much time for absorbing the information on the screen. Also notice that finding things in a database is most efficiently done by ordering and using locate if possible. Locate is fast and is pretty much independent of the database file size.

How do we judge the QLs performance against the other machines? I. QL much slower in its screen handling. This is evident in QUILL and ARCHIVE. I tried SPEEDSCREEN and LIGHTNING but they didn't improve the times. Some programs still manage to give fast screens. The EDITOR and TEXT87 are examples. Also the QL was multitasking QUILL while DBEasy was being run. That didn't affect the speed of doing the tasks because QUILL was sitting idle None of the other computers can multitask. None of the other computers can network either without expensive equipment. And the AT machines cost 2 to 5 times as much as the QL. At some point I will try these same tasks on a THOR and pass on the results. It should be much closer to the AT machines. If you were to judge the QL purely on response times it shows its age. If you judge it on overall features taking into account its powerful operating system, its cost, and how much you can accomplish with it then it still can hold its own against most PCs.

I have quite a group of QL Softwares and I'm developing more for both the QL and the IBM. All are priced low for what they do. I will list four of my newest. I invite you to call or write for a complete listing. Bill Cable.

WOOD AND WIND COMPUTING

SOFTWARE FOR THE QL and PC

QL SOFTWARE

CHAOS BUSTERS is a group of SUPERBASIC and ARCHIVE programs (DIRLUX, FILEX,DBEasy) that will help organize all your disks/microdrives so you can make sense out of them. It includes DIRlux which displays/prints directories is a column format and ahs a file finder. Also included is FILEX which makes a export file for ARCHIVE of your disks so you can make an instant database of all your files. You import the FILEX export file into ARCHIVE with DBEasy which is also included. DBEasy is an easy to use front end for ARCHIVE so you can make and manage all sorts of databases without knowing the first thing about ARCHIVE. DBEasy is also available separately and is described below. CHAOS BUSTERS is tied together through a Main Menu so you can do move around easily between the options. It is supplied with ample documentation plus extra software not described. If you already have DBEasy you can purchase CHAOS BUSTERS for $9.95. ($29.95 5 1/4 or 3 1/2 disk)

DBEasy : An ARCHIVE program that makes using ARCHIVE a snap. Everything is done from simple menus. You don't need to know any ARCHIVE commands but still have most of ARCHIVE's power. It allows for the creation and managing of databases for all applications. You can easily switch any of your databases. ($21.95 on 5 1/4 or 3 1/2 disk, $24.95 on 2 micro)

PC Software for IBM compatibles

PC-FOUR by PSION : The same QUILL, ABACUS, EASEL, and ARCHIVE that you use on your QL for the PC. They have more features than the QL version. They come on 5 1/4 360k disks with a thick manual and a tutor disk that make it easy to learn how to use them if you don't know already. I will include DBEasy free with any purchase of PC-FOUR.($99 on 5 1/4 disks)

DBEasy : for those who aready have PC-ARCHIVE I offer DBEasy in a PC version. It works almost exactlty like the QL version and you can export QL DBEasy database to the PC and back. ($21.95 on 5 1/4 or 3 1/2 disks)

Order today or write for a list of all software available :

Wood and Wind Computing
(Bill Cable)
RR-3 Box 92
Cornish, NH 03745
603 675 2218

SETTING UP QUILL FOR PRINTERS

Setting up TYPEFACES for F4

Using "INSTALL_BAS"

Quill Version 2.35 (or any)

The "INSTALL_BAS" program in our PSION suite is used to do all of these jobs to make our printer do what we want it to do. *INSTALL_BAS* is used to revise our *PRINTER_DAT* program in the PSION suite for use by the Quill, Easel, Archive, and Abacus softwares. So, how do we do it? Peter Hale gave one way to do it in the last issue of Update. This time we will do it in a simple fashion. If you had trouble with the procedures given in Peter's article, try this and when it works, go back to Peter's article for a more elaborate printer control set up.

FIRST A DEMONSTRATION:

This and the above style is gotten by use of the F4 Key and "B for BOLD"

This style is gotten by use of the F4 Key and "U for Underline".

This style is gotten by use of the F4 Key and "H" for High Script".

This style is gotten by use of the F4 Key and "L for Low Script"

Such Combinations *as these* three lines demonstrate *are gotten by using the* F4 Key and repeated electives of "B, H, L, or U".

But what about this style of print that comes up as "standard" when PRINT is elected at the Quill Command menu? Well, the normal "Printer_dat" program puts you in "draft quality mode", while this style is "Letter Quality" mode.

OK. The illustrations are just to show that the *quill F-4 Key* electives for *TYPEFACE* need not be for the print styles given to us in the PSION suite. In fact, each type of printer has its own typefaces that can be substituted for such PSION electives as *"high script, low* script, and bold. And, you can set in any style that you choose for these F4 electives.

The styles demonstrated are those that I elected to install in the Printer_Dat program for my KXP-1124 printer. Each user needs to decide what style is wanted and then proceed to change the Printer_dat program to his needs.

This is done by using the Printer Manual to find the Command Codes to "install" in the Printer Dat program. The "process of installing" is via the use of the "INSTALL_BAS" program that is in your Quill disk (MDV).

Now lets do it!

The electives given in the steps below are my own for my printer. Use your Printer Manual to substitute the command codes for the electives that you choose in each of the steps.

1. Leave FLP1 and MDV1 Empty and turn ON the QL. Then the F1 Key will put you in Super Basic.
2. We will use FLP commands but you can use MDV instead.
3. Remove the write protect tab on the floppy.
4. TYPE: *LRUN FLP1_INSTALL_BAS*
5. When the Install_bas program loads you will need to:
   a. elect the drive that your Quill disk is in.
   b. elect the ser port that you use for your printer.
6. The "Current Printer_Dat program will be loaded and this begins the process of installing your own printer codes.
7. A list of different Printers will be presented.
8. Use the Down Arrow key to highlight "OTHER".
9. Use the Right Arrow Key to erase OTHER and then type in the name of your printer.
10. Use the F2 Key to EDIT.
11. The printer_dat set-up will be displayed.
12. Use the down arrow key to elect each of the following lines to edit. As each line is highlighted, use the right arrow key to erase the line (if needed). Then type in the new line.
    a. PORT: Make this the Serial Port that you use for your printer.
    b. BAUD RATE: Should be 9600.
    c. PARITY: None.
    d. LINES: 66
    e. CHRS/LINE: 80 for Pica, 96 for Elite, or larger for other smaller styles of print that you want THIS Printer_Dat program to control.
    f. CONTINUOUS FORMS: YES.
    g. END OF LINE: CR,LF (carriage return and line feed).

Now we will begin to set in your SPECIAL PRINTER CODES to SET your desired functions.

h. PREAMBLE: This sets the printer when "PRINT" is elected with Quill or one of the other PSION softwares. I chose to set the following (command codes for KXP-1124)
27,120,1,27,107,2 (This sets LETTER QUALITY and then Courier Font, and then ELITE STYLE).
Refer to step "e" above. For this ELITE style the CHRS/LINE setting is 96.

i. POSTAMBLE: Here we want to assure that when PRINTING has been done, the FORM is fed up to the start of the next form. Insert:
FF,CR (Form Feed and Carriage Return)

j. BOLD ON: The Printer Codes for the Type

Style that you want the F4 Key to control: My choice was:

27,119,1,27,69,1 (the first three sets DOUBLE HIGH. The second three sets EMPHASISE MODE (double strike).)

k. BOLD OFF: Use the OFF codes for each of the ON codes used in j above. In my case the OFF Codes were: 27,119,0,27,70

l. UNDERLINE ON: My printer codes used for Underline ON was: 27,45,1. I decided that if I wanted to Underline something, I'd also shift into EMPHASISE MODE, 27,69,1. So the six commands are entered as: 27,45,1,27,69,1

m. Use the OFF codes for each of the ON codes in l above. (27,45,0,27,70)

n. HIGH SCRIPT ON: I decided to use my printer's Script font, which is gotten with the codes: 27,107,4.

o. The off codes for the above is to return to the Courier font, which is: 27,107,2

p. LOW SCRIPT ON: I decided to use my printer's ITALIC style and DOUBLE STRIKE, the command codes: 27,52,27,71

q. Off Codes for the above are: 27,53,27,72

r. This ends the Printer_Dat set-up, and none of the other lines were changed.

13. NOW WE NEED TO DO THREE MORE THINGS: I'LL STEP IT OUT A, B, C.

A. Run the cursor up to the NAME (first line).

B. Touch F2 to SAVE. Your new Printer_Dat program will be SAVED to disk or mdv,

C. Touch F5 to INSTALL. Your PRINTER_DAT program will be Installed as the one to use when PRINT is elected by one of the PSION programs.

Then lets do it again!

WHY? Well, I'll give you some of my illogical reasoning. *I like to keep things veery simple.* Instead of putting in a lot of special key codes, you'll now have ONE disk or mdv for ELITE STYLE. The F4 Keys are set up to give you the examples used in this article.

Next, REPEAT the steps given for setting up another Quill Disk for PICA 80 chrs/line, and still another for "another style". I set up my third disk for the nice MICRON 120 chrs/line that my KXP-1124 has. To do these two additional disks only required changing two lines, the line for CHRS/LINE and the Preamble line for the SET-UP UPON BOOT IN.

There are several reasons that I decided to use separate disks for the three styles. First, the PSION suite does not let you have different "DESIGN" for successive printed pages. Then, the different widths of print (Pica, Elite, Micron) mixed in printed lines would cause FORMAT inconsistancies. Even so, there is one slight problem with this Printer_dat. When using BOLD, the characters are DOUBLE HIGH. For the best results BOLD should be used only for HEADERS, and for printing on the FIRST line of a separated paragraph. Otherwise when BOLD is used for more than one line there isn't any separation between the two lines.

YOU COULD SET UP LESS SPACING BETWEEN LINES

One could set up other PRINTER_DAT programs to use CONDENSED PRINT or SUPERSCRIPT for printing. These styles used for manuscript printing need shorter spacing between lines. Since I'm not a Lawyer and do not need to use "fine print" to hide legalese in contracts, I really do not need such small printing.

But if you do, you can use the PREAMBLE line to SET the Print style, then superscript, and THEN, SET the PRINTER LINE SPACING to something other than 1/8" line spacing, say 1/16" for superscript, or 5/64" for condensed. Those settings would give about 120, or 96 lines per page form, and a whale of a lot of data per page. Imagine, a print style of MICRON 120 chrs/line, then superscript, then 1/16" line spacing, 1/16", to give 120 lines per page, to give over 12000 characters per printed page! Or, Condensed 160 Chrs/line, superscript, 1/32" line spacing to give 240 lines per page, to give 40,000 characters per page! (My math could be off.)

Summary

1. Every user should learn to use the *INSTALL_BAS* program to make tailored *PRINTER_DAT* set-ups that your printer is capable of supporting.

2. Once you do a simple new set-up with the Install_bas program, you'll remember how and you can always revise the _dat program to do whatever you want to do with your printer.

3. Each brand of printer has its own best print styles that should be taken advantage of with custom Printer_dat set-ups.

4. While I cannot find much use for setting up different styles of print that have different character widths, it can be done. One can account for inconsistant line lengths of separated paragraphs, but mixed widths of print in the same line would be difficult to format. Also, the DESIGN FUNCTION would not provide consistant page length with Mixed styles and different printer line spacing.

5. A simple approach is to use different disks/mdvs for Print styles, with all styles on one disk's printer dat having the same character width.

6. If you DONT learn to use Install_bas, you will not have much lattitude of style selections. You will ALWAYS: Print in Draft Quality, Pica 80 Chr lines, and your F4 Typeface electives will be for superscript and subscript styles. These are fixed by the given Printer_dat program.

learn to use INSTALL_BAS. To do so just requires a little practice, and coordination with the printer manual. *EnJoY! _BJ-*

The following article, by David McCall, is second in a series. The first article was given in the October 1989 issue. The Program Listing given in this issue can be linked with that given in the Oct. issue. Or, the individual programs will work independantly.

WINDOW DEFINING:
A Program, part III

by David McCall

This is the next to last article for the window defining program. The first one described the machine language routine which was necessary to swap screens. The second one detailed the variety of functions needed and now the procedures will be discuss. After this series has been completed you should have a better understanding of SuperBASIC's functions and procedures and their variety of uses along with a better understanding of the value of using machine code to supplement a programs speed and versatility.

As in the previous articles indentation has been used throughout the writing of these procedures. Most of the procedures are straight forward but different techniques have been used to illustrate the power and versatility of SuperBASIC especially when ToolKit II is installed. Most of the procedures use the Toolkit's additional utilities, PRINT USING and CHAR INC. The procedures can be rewritten to avoid these routines if it is necessary.

Let's start off by describing the first procedure at line 4000. The procedure is called **done** and takes four parameters **x%, y%, len%,** and **value%.** The procedure prints the **value%** in a signed field of length, **len%,** at position **x%,y%** and clears the default value which was printed at **x%+1,y%.** This procedure is used after a value has been typed in to update the screen.

PRINT USING was used to ensure the number lined up properly and stayed within its allotted space. The PRINT USING function takes a string as its first parameter, which describes how the successive parameters will be printed. In our case the string was made up of a SuperBASIC function FILL$ which creates a string of character of a desired length. A '-' was concatenated before it to ensure there was room for the sign to be printed. For more information on the above functions read the Toolkit II manual section 13.1 and the SuperBASIC manual section KEYWORDS page 23.

The next procedures **print_list_max_chr**, line 4050, and **print_list_max_point** line 4110, displays the maximum number of characters that can be displayed in the window (**max_xc,max_yc**) and the maximum point positions (**max_xp,max_yp**), the graphics origin (**xg,yg**), the number of pixels which corresponds to the window height, and the midway point between the origin and the maximum point position. Arrows are printed after each value to indicate which direction it refers to. This has proved useful to clear up the confusion about which direction the x and y parameters refer to in different commands.

Remember **max_xc, max_yc, max_xp,** and **max_yp** are the functions written in the previous article. The global variables are:

scl = the SCALE of the window
xg,yg = the graphics origin defined by the SCALE command
scr_x,scr_y = the x and y dimension of the window
brder = the BORDER width
cxi,cyi = the number of pixels a character occupies in the x and y directions
SWAP,swap2 = the machine language address to swap the command and defining windows, respectively
xp0,yp0,sx0,sy0 = the position and size of the window being swapped
cb1,cb2,cb3 = the color of the BORDER
cp1,cp2,cp3 = the color of the PAPER
ci1,ci2,ci3 = the color of the INK
csw,csh = character size width and height
doit = the flag which indicates what to display in the defining window after a command
a_or_m = the flag which indicates whether the window is being moved or altered in size

The procedure that does most of the work is **recalculate**, line 4220. It takes care of erasing the old window, putting the new one back up, and updating the appropriate variables. Here we see the first use of the machine language routine which was loaded into the location pointed to by **swap2**. As you remember the parameters for this routine are save/recall, origin, and size, see line 4230. If the save/recall position is one the previous screen is recalled which is the first thing that is done in this procedure. Next it checks to see if the window size is too big and adjusts if necessary. The new window is opened and set up. Next the origin and size for our machine language swap routine are recalculated, see part I of this series and lines 4310 to 4340. The screen is save under our new window and then the window is displayed.

The next two procedures displays information to the user. The first, **results** line 4400, displays all the relevant information about the current window, the defining window, that is being constructed. The second one, **commands** line 4690, prints a table of the commands that are available. Everything is labelled, placed and colored to ease in reading. An interesting approach has been used to print this information. One color of text is printed at a time and each time the ink color is change the cursor is repositioned at the top and the next mask is printed. This is accomplished by using OVER 1 and tends to save some memory by not repeatedly changing the INK color over and over again.

The **display** procedure, line 4790, is an added visual feature which as proved useful in creating windows. It fills the current window with 'QL' or puts a hash at the graphics origin depending on the last command executed. This helps size the window when you want your characters to exactly fit the window or when you want the origin in a specific location. By looking at lines 4800 to 4870, we see that the SELect ON command was used instead of the IF-THEN-ELSE construct. This decision was made in part to show an alternative to the IF statements and to allow an easier change in the future if different displays are wanted after other commands.

Of course the window to display the information, the command window, needs to be created. This is accomplished in the procedure **setup**, line 4900. LOCal variables are used with same name as global ones. This allows us to keep the same descriptive variable names without changing the global values. The window parameters are READ in through DATA lines, line 2 to 5. (NOTE, lines 8 to 13 are the initial values of the defining window and will be used in the main program.) This allows easier changes to the window without searching through the entire program to find where the window was defined. A RESTORE should always be used before READing any DATA because the line number from where the DATA will start to be READ is not reinitialed with the RUN command. The text '>> MOVE <<' or '>> ALTER <<' is also displayed depending upon the value of **a or m** which will be set in the main program. Finally, the area of the screen which the command window will occupy is saved.

Last but not least, a procedure to print the title and author of the program is included, **name** line 5120. The extra underlined spaces are used to draw a line to separate the title from the rest of the command window and underline the author's name. In some cases UNDERlining or drawing a BLOCK for a straight line is faster and easier.

The line numbering used has been constructed to ensure that all the articles can be merged together without out conflict. In the final article the simple main program will be discussed. The functions and procedures presented in this series is a basis to build and learn from. In some instances a better approach could have been used but the intent was to show several different ways to attack a problem. If you don't want to type in the listing an "Issue Disk" will be available.

LISTING A

```
1  :          REMark Command window
2  DATA 301,211,0,0:          REMark screen - height,width,x,y
3  DATA 0,0,0:                REMark paper color, contrast, stipple
4  DATA 2,2,0:                REMark ink color, contrast,stipple
5  DATA 2,0,12,10:            REMark char size - width,height,x_inc,y_inc
6  :
7  :          REMark Defining window
8  DATA 100,50,320,128:       REMark screen - height,width,x,y
9  DATA 1,7,7,0:              REMark border size, color, contrast,stipple
```

```
10 DATA 0,0,0:                  REMark paper color, contrast, stipple
11 DATA 2,2,0:                  REMark ink color, contrast,stipple
12 DATA 100,0,0:                REMark scale, (x,y) graphics origin
13 DATA 2,0,12,10:              REMark char size - width,height,x_inc,y_inc
14 :
4000 DEFine PROCedure done(x%,y%,len%,value)
4010    INK#3,4
4020    AT#3,x%,y%: PRINT USING#3, '-'&FILL$('#',len%),value:AT#3,x%+1,y%
        :PRINT#3,FILL$(' ',len%+1)
4030 END DEFine done
4040 :
4050 DEFine PROCedure print_list_max_chr
4060    AT#3,8,13
4070    INK#3,4:PRINT USING#3,'## ↑↓',max_xc
4080    AT#3,9,13:PRINT USING#3,'## ↔',max_yc
4090 END DEFine print_list_max_chr
4100 :
4110 DEFine PROCedure print_list_max_point
4120    AT#3,8,13:
4130    INK#3,4:PRINT USING#3,'-### ↔',max_xp
4140    AT#3,9,13:PRINT USING#3,'-### ↑↓',max_yp
4150    AT#3,12,13:PRINT USING#3,'-### ↔',xg
4160    AT#3,13,13:PRINT USING#3,'-### ↑↓',yg
4170    AT#3,16,12:PRINT#3,scr_y-2*brder-1
4180    PRINT USING#3,'\   -###.## , -###.##',-(max_xp-xg)/2,-(max_yp-yg)/2
4190    name
4200 END DEFine print_list_max_point
4210 :
4220 DEFine PROCedure recalculate
4230    CALL swap2,1,xp0,yp0,sx0,sy0
4240    IF scr_x+xp>512 THEN scr_x=512-xp
4250    IF scr_y+yp>256 THEN scr_y=256-yp
4260    OPEN#4,'scr_'&scr_x&'x'&scr_y&'a'&xp&'x'&yp
4270    PAPER#4,cp1,cp2,cp3:INK#4,ci1,ci2,ci3
4280    SCALE#4,sc1,xg,yg
4290    CSIZE#4,csw,csh
4300    CHAR_INC #4,cxi,cyi
4310    xp0=INT(xp/8-.5):IF xp0<0 THEN xp0=0
4320    sx0=INT(scr_x/8+.5)+2:IF xp0+sx0>64 THEN sx0=64-xp0
4330    yp0=yp-1:IF yp0<0 THEN yp0=0
4340    sy0=scr_y+1:IF yp0+sy0>512 THEN sy0=512-yp0
4350    CALL swap2,0,xp0,yp0,sx0,sy0
4360    BORDER#4, brder,cb1,cb2,cb3
4370    CLS#4
4380 END DEFine recalculate
4390 :
4400 DEFine PROCedure results
4410    OVER#3,1
4420    INK#3,4:PRINT#3,'scr_   x   a   x'
4430    PRINT#3,'  BORDER   , , '
4440    PRINT#3,'   PAPER   , ,'
4450    PRINT#3,'     INK   , ,'
4460    PRINT#3,'   SCALE   ,   ,'
4470    PRINT#3,'   CSIZE   ,'
4480    PRINT#3,'CHAR_INC   ,';
4490    PRINT#3,\\'plot: max_x ='\TO  6;'max_y ='\\TO 6;'min_x ='\TO 6;'min_y ='
4500    PRINT#3,\'print: max_x = '\TO 7;'max_y ='
4510    PRINT#3,\'pixel: max_x ='\TO 7;'max_y ='
4520    AT#3,0,0
4530    INK#3,7:PRINT USING#3,'    ### ### ### ###\',scr_x,scr_y,xp,yp
4540    PRINT USING#3,'         ## # # #\',brder,cb1,cb2,cb3
4550    PRINT USING#3,'         # # #\',cp1,cp2,cp3
4560    PRINT USING#3,'         # # #\',ci1,ci2,ci3
4570    PRINT USING#3,'         ### -### -###\',sc1,xg,yg
4580    PRINT USING#3,'         # #\',csw,csh
4590    PRINT USING#3,'         ## ##\',cxi,cyi
4600    PRINT USING#3,'\            -####.### ↔\             -### ↑↓\'
        ,max_xp,max_yp
4610    PRINT USING#3,'\           -### ↔\            -### ↑↓\',xg,yg
4620    PRINT USING#3,'\            ## ↑↓\             ## ↔\',max_xc,max_yc
4630    PRINT USING#3,'\            ### ↔\             ### ↑↓\'
        ,max_xpix,max_ypix
4640    INK#3,2:UNDER#3,1
4650    AT#3,18,0:PRINT#3,'                        ';
```

```
4660    OVER#3,0:UNDER#3,0
4670 END DEFine results
4680 :
4690 DEFine PROCedure commands
4700    OVER#3,1
4710    INK#3,7:PRINT#3,'    M'\'    A'\'    L'\'    S'\'    I'\'    P'\'    B'\
        '    C'\'    R'\'    A'\'    S'\'    A'\'    S'\'    M'\'⇔↑↓'\\'SPC'\
        ' RTN'\'      ESC'
4720    AT#3,0,0
4730    INK#3,4:PRINT#3,'     ove'\'     lter'\'     etter size'\'     cale'\
        '     nk'\'     aper'\'     order'\'     ommands'\'     esults'\
        '     bsolute position'\'     tep move'\'     absolute size'\
        '     tep alter'\'     ode'\'     -moves screen or'\'     alters size'\
        '    -redraws screen'\'&'
4740    AT#3,9,0
4750    INK#3,2:PRINT#3,'ctrl'\'ctrl'\' alt'\' alt'\'ctrl'\\\\\'          EXITS'
4760    OVER#3,0
4770 END DEFine
4780 :
4790 DEFine PROCedure display
4800    SELect ON doit
4810       ON doit=1
4820          FOR i=1 TO max_xc
4830             PRINT#4,FILL$('QL',max_yc)
4840          END FOR i
4850       ON doit=2
4860          LINE#4, 5,0 TO -5,0:LINE #4,0,-5 TO 0,5
4870    END SELect
4880 END DEFine display
4890 :
4900 DEFine PROCedure setup
4910 LOCal scr_x,scr_y,xp,yp,cp1,cp2,cp3,ci1,ci2,ci3,csw,csh,cxi,cyi
4920    RESTORE 4000
4930    READ scr_x,scr_y,xp,yp
4940    READ cp1,cp2,cp3
4950    READ ci1,ci2,ci3
4960    READ csw,csh,cxi,cyi
4970    IF scr_x+xp>512 THEN scr_x=512-xp
4980    IF scr_y+yp>256 THEN scr_y=256-yp
4990    OPEN#3,'con_'&scr_x&'x'&scr_y&'a'&xp&'x'&yp&'_10'
5000    PAPER#3,cp1,cp2,cp3:INK#3,ci1,ci2,ci3
5010    CSIZE#3,csw,csh
5020    CHAR_INC #3,cxi,cyi
5030    AT#3,0,0
5040    IF a_or_m THEN
5050       PRINT#3,'>> MOVE << '
5060    ELSE
5070       PRINT#3,'>> ALTER <<'
5080    END IF
5090    CALL SWAP,0,0,0,38,211
5100 END DEFine setup
5110 :
5120 DEFine PROCedure name
5130    INK#3,4
5140    AT#3,19,6:PRINT#3,'Wind_Def'
5150    PRINT#3,'by  ';
5160    INK#3,7
5170    PRINT#3,'David McCall  ';
5180    INK#3,4
5190    PRINT#3,'©1989'
5200    INK#3,2:UNDER#3,1:OVER#3,1
5210    AT#3,18,0:PRINT#3,'                    ';
5220    AT#3,20,4:PRINT#3,'            ';
5230    UNDER#3,0:OVER#3,0
5240 END DEFine name
```

A QL SPREADSHEET MATRIX --- Bob Hartung

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1								1990 FINANCIAL REPORT							
2															ANNUAL
3	EXPENSE ITEMS:		JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	TOTALS
4															
5		Bus.miles	0	0	0	0	0	0	0	0	0	0	0	0	sum(C5:N5)
6	IRS mileage		C5*0.255	--	--	--	--	--	--	--	--	--	--	--	--
7	Postage & fees		0	0	0	0	0	0	0	0	0	0	0	0	--
8	. . . .		0	0	0	0	0	0	0	0	0	0	0	0	--
9			------	------	------	------	------	------	------	------	------	------	------	------	------
10	EXPENSE/MONTH		sum(C6:C8)	--	--	--	--	--	--	--	--	--	--	--	sum(C10:N10)
11															
12	INCOME SOURCES:														
13															
14	Widget Co.		0	0	0	0	0	0	0	0	0	0	0	0	sum(C14:N14)
15	ABC Specialties		0	0	0	0	0	0	0	0	0	0	0	0	--
16	. . . .		0	0	0	0	0	0	0	0	0	0	0	0	--
17			------	------	------	------	------	------	------	------	------	------	------	------	------
18	INCOME/MONTH		sum(C14:C16)	--	--	--	--	--	--	--	--	--	--	--	sum(C18:N18)
19															
20	1990 BEGINNING BAL		(enter constant)												
21															
22	TOTAL INCOME		O18												
23			-------												
24	CASH ON HAND		C20+C22												
25															
26	TOTAL EXPENSES		O10												
27			-------												
28	ENDING BALANCE		C24-C26												
29			""												
30			""												

After creating and saving a copy of the above Abacus (SS) matrix for a simple accounting format, only a few additional steps are required to adapt it for any similar uses of your own. It is easier to center the heading in row 1 if it is done last so start at cell A3 after booting up SS. The default cell-width on boot-up is 10, but we will change this to 8 later on so most printers will do 144-column printouts of the full spreadsheet in one pass.

Text entries are made by keying " as the very first character. Text entries made to one cell may span two cells as long as the second cell is empty. The closing " is supplied by SS when ENTER is keyed. To avoid confusion, if you miss entering the opening " it is best to key ESC and start over. Enter the monthly headings across row 3, and all the other labels. If the left-most labels in column A all fill the A cells they will retain a uniform left margin when a right-justify is used for text. Do not enter the ------ lines nor the formulas just yet as printed out in the example or indicated by two -- dashes.

When you have completed all the labeling with ENDING BALANCE as cell A28, continue adding empty cells by entering a single " in column A for as many expense and income catagories as you intend to add, plus a few extras for any you may want to add later. The reason for this is that when additional rows are inserted anywhere above the final row in the matrix, anything in the last lines is overwritten, so nothing is lost if these are null cells.

Change the cell-width by keying F3, G for Grid, W for Width, 8 for the new column width, FROM A TO O. Set text to right-justify by keying F3, J for Justify, key ENTER to change CELLS, ENTER for TEXT, R for Right, and range C3:O28. Enter whatever heading you want to use in row 1.

We are ready now to start plugging in formulas and numbers. Move the cursor to C5 and enter 0 (zero). No opening quote " is used for numeric or formula entries. To repeat this across row 5 the ECHO command is used by keying F3, E for Echo, ENTER (for the cell on which the cursor is resting), range C5:N5. Do this for rows 7, 8, 14, 15, 16. After entering "------ as text in cells C9 and C17, use ECHO to repeat in ranges C9:O9 and C17:O17, respectively. If you will have 5-digit figures in the O column, you may want to use 8 dashes there for neatness.

Enter the formula as given in cell C6 and ECHO this in range C6:N6. Enter the formula as given in C10 and ECHO this in range C10:N10. Enter the formula as given in C18 and ECHO this in range C18:N18. Enter the formula as given in O5 and ECHO this in range O5:O8. Enter the formula as given in O14 and ECHO this in range O14:O16. In cell O10 enter sum(C10:N10), in cell O18 enter sum(C18: N18), in cell C20 enter 0 or a beginning balance, in cell C22 enter O18, in cell 23 enter "--------, in cell 24 enter C20+C22, in cell 26 enter O10, in cell C27 enter "-------, in cell C28 enter C24-C26. CELL formats persist till changed.

Change all numbers to right-justify by keying F3, J for Justify, D to change defaults, N for numbers, R for Right. If any cells need to be moved right after new entries, this may be done by the CELL mode of the Justify sequence, then specifying the range or just a single cell, eg: C8.

To create a two-place decimal format for aligning all monetary figures, key F3, U for Units, D for Default, D for Decimal, and two for number of places. For better clarity, any single cell or range of cells such as mileage may initially be set to a single 0 with CELL and GENERAL, then reset to two-place decimals as entries are made. For a window display, with cursor at C1 key F3, W, V, ENTER.

With this matrix as given, a narrow carriage (letter-width) printer that has a condensed font of 17 c.p.i. will print the entire spreadsheet in one pass if the column-width is set to 144 by using the DESIGN command. If your printer will not do 144 characters per line or you wish to do printouts in a larger font, then the spreadsheet may be done in two or more segments, eg: key F3, P for Print, ENTER for display, N for grid only, range A1:J28 for the first segment, then repeat, using K1:O28 as the range.

The matrix we now have may be saved and used as the foundation for any spreadsheets that use this type format. To insert more rows into it, with the cursor on the row following the position where new rows are to be added, key F3, G for Grid, I for Insert, ENTER (or row number), and the number of rows to be inserted. ECHO any values or formulas as necessary. Any existing formulas are adjusted when rows are re-numbered by using Insert. Add null "" rows to the bottom of the matrix if necessary.

TS-2068 BUSINESS SOFTWARE

AR1-9.B1

By: Jim Bretz, Pernam, MN.

Manages Customer Charge Accounts
Posts new sales
Computes interest charges
Posts payments to accounts
Sums accounts balances
Prints out Reports
Prints out invoices

This comprehensive software proves that the TS-2068 is a mighty fine computer for running a business. The equilivant of this program in use throughout the World costs hundreds of dollars. Jim Bretz designed and uses this AR1-9 software to run his large Hardware business. Jim has versions of the software for both LKDOS and SPDOS (Millenia K DOS). The version given here is for LKDOS and the V-3 LKDOS Printer Driver in EPROM. However, any printer driver that supports TAB can be used. Also, the software requires the 64 column screen to be enabled. This can be via use of TASWIDE or the Zebra OS-64 Cartridge. If you are using LKDOS then TASWIDE would be appropriate. Incidentally if you dont have TASWIDE, which works great with LKDOS, you should spend a few bucks to get it (RMG Enterprises is a good source.).

If you are using a DOS other than LKDOS (or Cassette), then as you key in the program just substitute the LOAD and SAVE commands for the storage medium that you use.

The Software programming is done entirely in BASIC which makes the operation very user friendly and adaptable for customizing to your special needs. In fact, it is so user friendly that only a few tips to get started need to be discussed. The Main Menu is given below:

ACCTS REC. Main Menu

1. ENTER NEW FILE
2. VIEW FILE
3. POST SLIPS
4. COPY TO PAPER
5. REPORTS
6. NEW HEADER/CLOSING
7. EDIT LINE
8. CLOSE FILE/END DAY
9. SAVE DATA (BACK UP)

Key in the whole program and SAVE with a starting line of 9500. (SAVE "AR1-9.B1" LINE 9500). Next SAVE a couple of dummy account files that will be needed. (DIM a$(20,5,21): SAVE "01001a.A$"DATA a$()).

And: (DIM d$(20,24,20): SAVE "01001d.A$" DATA d$()).

These two character arrays saved to disk will become dummy files for the program to sense a need to establish new files.

The Auto Load program

After Saving the main program and the two dummy files, key in the LKDOS Autostart program. This Autostart program may need slight changes to initialize your own CPI interface with POKES to the LKDOS. Lines 30 and 40 assumes that you have SAVED the taswid Code via <SAVE "taswid.C1" CODE 63223,1492 > to the same disk as you saved the main program. Key in the Autostart program with your needed changes and Auto SAVE it by: < GO TO 60 >. Line 60 lowers RAM TOP to 300 bytes above the program length to conserve disk space. Then the Autostart Program itself will restore FREE Memory as it loads in (line #5) and sets RAM TOP to account for the taswid code load.

SOME MAIN PROGRAM TIPS

When the main program loads (via the Autostart program) it begins running at line #9500. Line #9530 prompts for input of DATE. The date must be SIX numbers as: 010190, which would represent the date of Janurary 1, 1990.

Then Line #9560 loads in the first account. In this case it is the dummy account that you saved with the name "01001a.A$". Then line #9565 loads in the second part of the account which you saved with the name "01001d.A$".

Then when these two files (dummys) are loaded, the Main Menu is presented.

Select <1> ENTER NEW FILE. Now here is a little quirk. The file number that you enter must be 7 characters in length and must conform closely to the dummy file name, which is only 6 characters in length. Files must begin with the first two digits "01" and the last two digits "1a" through "1z". EXAMPLE: 010001f would be accepted, but 020002a would not. A close examination of the program lines beginning at line #410 will be helpful to understand the composition of the "account number" requirements.

Enter several new accounts and input dummy figures for sales by category. Then use the main menu to recall these accounts. After a bit of practice the organization scheme of the accounts will become familiar.

Since this is such a comprehensive software and does so many functions, there will be a feed back to answer the users questions. If you should have problems, send your questions to Update Magazine and they will be forwarded to Jim Bretz for his answers. Please include two Stamped envelopes, one a SASE. But lets not call Jim for problem answers as he is a business man who goes to work every day and needs his sleep.

This Business software is available as an Issue disk, ready to use for LKDOS.

```
VARIABLES:
    Am---Val D$(x,z,13 to 20)
    Amt--Amount total Debit
    Bal--Amount runing Balance
    Crt--Amount total Credit
    D----Input for description menu
    Des--Description, Val D$(x,z,11 to 12)
    Int--Amount total Interest
    K----Length of S$
    Ln---Line posistion, PEEK 23689 (SPOSN)
    Lb---Used in Lbl
    Lbl--Over due Balance
    P----For/Next loop counter
    Q----For/Next loop counter, printer
    S----Used to justify R$ and S$
    T----Terms flag
    Tbal-Gross total Balance,
    Tlbl-Gross Over due Balance
    Tmi--Total months Interest added
    Tint-Gross total Interest
    X----Account number, Val X$(6 to 7)
    Z----For/Next loop counter
```

```
  1 REM  Accounts Rec. T/S 2068          1986 Jim
Bretz
 304 PRINT CHR$ 2
 305 PRINT AT 1,o;X$'A$(X,1)'A$(X,2)'A$(X,3)'A$(X,4,
1 TO 14);" ";A$(X,5,1 TO 2);".";A$(X,5,3 TO 7)
 307 IF A$(X,1,1 TO 4)="    " THEN  PRINT "FILE NOT
FOUND": PAUSE o: GO TO j5
 309 RANDOMIZE USR 64300: PRINT CHR$ 3
 310 PRINT AT 1,48;"Sheet:";A$(x,5,12 TO 13);AT 2,49
;"LSmt:";A$(X,5,8 TO 9);"/";A$(X,5,10 TO 11);AT 3,49
;"Date:";g$;AT 4,48;"Terms:";T$(VAL A$(X,5,14))
 320 PRINT AT 7-(z=16),o;"No! Date! Slip !Descriptio
n !Credit !Interest! Amount ! Balance "
 349 RETURN
 352 LET s=VAL D$(x,z,13 TO 20)
 354 LET k=9
 355 LET r$=STR$ s: LET l=LEN r$
 360 IF L(2 THEN  LET R$="0"+R$: LET L=2
 365 DIM s$(k): LET s$(k-2)=".": LET s$(k-1 TO k)=r$
(l-1 TO l): LET s$(k-l+(k=l) TO k-3)=r$(1 TO l-2): R
ETURN
 400 REM leagel acct sub
 410 INPUT "ACCT. NO.";Y$: IF LEN Y$()7 THEN  GO TO
502
 415 POKE 23658,0: LET x$=Y$: LET X=VAL X$(6 TO 7):
IF x)20 THEN  GO TO 502
 420 IF X$(1 TO 2)()C$(1 TO 2) THEN  PRINT "PLEASE I
NSERT DISC NO. ";X$(1 TO 2)"AND TRY AGAIN.": PAUSE
0: GO TO 500
 425 IF VAL x$(3 TO 5)>48 THEN  GO TO 502
 430 IF X$(3 TO 5)()C$(3 TO 5) THEN  GO TO 450
 440 RETURN
 450 RANDOMIZE USR 100: SAVE c$+"a.A$" DATA a$( )
 452 RANDOMIZE USR 100: SAVE c$+"d.A$" DATA d$( )
 453 IF z$)"7" THEN  RETURN
 454 RANDOMIZE USR 100: LOAD X$(1 TO 5)+"a.A$" DATA
a$( )
 456 RANDOMIZE USR 100: LOAD X$(1 TO 5)+"d.A$" DATA
d$( )
 458 LET c$=x$(1 TO 5)
 499 RETURN
 501 GO TO 505
 502 BEEP .1,1: BEEP 1,.1
 505 BEEP .1,1: BEEP .1,5: BEEP .1,9
 510 CLS : PRINT "ACCTS REC. MAIN MENU:";AT 2,5;"1.
ENTER NEW FILE",TAB 5;"2. VIEW FILE",TAB 5;"3. POST
SLIPS",TAB 5;"4. COPY TO PAPER",TAB 5;"5. REPORTS",T
AB 5;"6. NEW HEADER/CLOSING",TAB 5;"7. EDIT LINE",TA
B 5;"8. CLOSE FILE/END DAY",TAB 5;"9. SAVE DATA (Bac
k-Up)"
 520 INPUT Z$: IF CODE Z$)48 AND CODE Z$(58 THEN : C
LS : GO TO VAL Z$*1000
 530 GO TO j6
 549 CLEAR : RANDOMIZE USR 100: SAVE "AR1-9.B1" LINE
 9500
1000 CLS : FLASH 1: PRINT AT o,9;"ENTER NEW FILE"'':
 FLASH o: GO SUB j4
1010 LET X=VAL X$(6 TO 7)
1020 IF CODE A$(X,1)()32 THEN  GO TO 1900
1025 INPUT "ACCT. NAME ";A$(X,1): PRINT A$(X,1)
1030 INPUT "St. ADDRESS";A$(X,2): PRINT A$(X,2)
1032 INPUT "RR or BOX";A$(X,3): PRINT A$(X,3)
1035 INPUT "CITY";A$(X,4): PRINT A$(X,4)
1040 INPUT "STATE, & ZIP";A$(X,5,1 TO 7): PRINT A$(X
,5,1 TO 7)
1050 POKE cap,8: INPUT "IS THIS CORRECT Y/N ?";Y$: I
F Y$()"Y" THEN  CLS : PRINT "ACCT NO. ";X: POKE cap,
o: GO TO 1025
1110 CLS : PRINT "TERMS MENU:": PRINT AT 2,5;"1. ";T
$(1)'TAB 5;"2. ";T$(2)'TAB 5;"3. ";T$(3)'TAB 5;"4. "
;T$(4)'TAB 5;"5. ";T$(5)
1120 INPUT A$(X,5,14): IF CODE A$(X,5,14))48 AND COD
E A$(X,5,14)(54 THEN  GO TO 1130
1125 BEEP .1,1: BEEP 1,.1: GO TO 1110
1130 LET A$(x,5,12 TO 13)="01"
1140 CLS : INPUT "Date of LSmt.",A$(X,5,8 TO 11): PR
INT "LSmt: ";;A$(x,5,8 TO 9);"/";A$(x,5,10 TO 11): L
ET D$(X,1,1 TO 4)=A$(X,5,8 TO 11)
1145 LET D$(x,1,5 TO 10)="   00": LET D$(x,1,11 TO 1
2)=" 1"
1150 INPUT "Enter beginning Balance";Y$: LET D$(x,1,
13 TO 20)=Y$: IF LEN Y$(1 THEN  GO TO 1150
1151 FOR z=1 TO LEN Y$: IF CODE Y$(z))57 THEN  GO TO
 1150
1152 IF CODE y$(z)(48 THEN  GO TO 1150
1153 NEXT z
1155 INPUT "Enter OVER DUE Balance";y$: LET a$(x,5,1
5 TO 21)=y$: IF LEN y$(1 THEN  GO TO 1155
1156 FOR z=1 TO LEN y$: IF CODE y$(z))57 THEN  GO TO
 1155
1157 IF CODE y$(z)(48 THEN  GO TO 1155
1158 NEXT z
1159 LET Z=1: GO SUB j1: LET amt=s: PRINT "Balance F
or. ";s$: IF s(=0 THEN  LET D$(x,1,11)="0": GO TO 12
50
1160 IF S)0 THEN  LET D$(x,1,11)="1": INPUT "Enter b
eginning Interest";Y$: LET D$(x,1,5 TO 10)=Y$: IF LE
N y$(1 THEN  GO TO 1160
1161 FOR z=1 TO LEN y$: IF CODE y$(z))57 THEN  GO TO
 1160
1162 IF CODE y$(z)(48 THEN  GO TO 1160
1163 NEXT z
1164 LET z=1: LET s=VAL D$(x,1,5 TO 10): LET k=7: GO
 SUB j3: PRINT "Interest charge";s$
1170 LET s=s+amt: GO SUB j2: PRINT "Total Due:  $";s
$
1250 PRINT AT 21,o;"Press Any Key for MENU""Press "
"C"" to Enter a new Acct.": POKE cap,o
1260 IF INKEY$="" THEN  GO TO 1260
1265 IF INKEY$="c" THEN  GO TO 1000
1275 GO TO j5
```

```
1900 PRINT "THIS ACCT ";A$(X,1)'"IS IN USE,"'' DO
YOU WISH TO DELETE ? Y/N": INPUT Y$: IF Y$()"Y" TH
EN  GO TO 1270
1904 PRINT : PRINT "This File ? ";A$(X,1);",(Y/N)"
: INPUT Y$: IF Y$="Y" THEN  LET A$(X,1,1 TO 4)="
  ": FOR Z=1 TO 24: LET D$(X,Z)="": NEXT Z: GO TO
505
1906 GO TO j6
2010 CLS : FLASH 1: PRINT AT o,12;"VIEW FILE": FLA
SH o
2015 GO SUB 3015
2505 POKE cap,8: PAUSE o: IF INKEY$="R" THEN  LET
Z=1: CLS : GO TO 3030
2510 GO TO 2000
2950 PAUSE o: GO TO 2000
3010 CLS : FLASH 1: PRINT AT o,10;"POST SLIPS": FL
ASH o
3015 LET Z=1
3020 GO SUB j4
3030 GO SUB j0
3045 POKE cap,8: INPUT "Correcrt File Y/N";Y$
3048 IF Y$="Y" THEN  GO TO 3050
3049 GO TO j6
3110 PRINT AT 8,z;z;TAB 3;D$(x,z,z TO 2);"/";D$(x,
z,3 TO 4);TAB 16;"Balance For.";
3120 IF VAL D$(x,z,13 TO 20)(0 THEN  GO SUB j1: PR
INT AT 8,27;S$;TAB 55;S$: LET amt=0: LET bal=s: LE
T crt=ABS s: LET int=0: LET in=int: GO TO 3200
3130 LET s=VAL D$(x,z,5 TO 10): LET k=7: GO SUB j3
: PRINT AT 8,38;s$: LET int=s: LET in=int
3134 GO SUB j1: PRINT AT 8,45;s$;TAB 55;s$: LET am
t=s: LET bal=s: LET crt=0
3202 FOR z=2 TO 24
3205 IF z=16 THEN  INPUT "scroll?";Y$: PRINT AT o,
o;: FOR h=1 TO 10: RANDOMIZE USR 2361: NEXT h: FOR
 h=1 TO 7: PRINT "
                                    ": NEXT h: GO SUB
j0: PRINT AT ln-9,o;
3210 IF D$(x,z,1)=" " THEN  GO TO 3300
3215 LET des=VAL D$(x,z,11 TO 12)
3220 PRINT TAB 1-(z)9);z;TAB 3;D$(x,z,1 TO 2);"/";
D$(x,z,3 TO 4);TAB 9;D$(x,z,5 TO 10);TAB 16;I$(des
);: LET ln=24-PEEK 23689
3225 IF des(8 THEN  GO SUB j1: PRINT TAB 45;s$;: L
ET amt=amt+s: LET bal=bal+s: LET s=bal: GO SUB j2:
 PRINT TAB 55;s$: NEXT z: GO TO 3400
3230 IF des(10 THEN  LET am=VAL D$(x,z,13 TO 20):
GO SUB j1: PRINT AT ln,27;s$;: LET s=in: LET k=7:
GO SUB j3: PRINT TAB 38;s$;: LET s=(am-s)*(am-s)=0
): LET in=(in-am)*(in-am)=0): GO SUB j2: PRINT AT
ln,45;s$;: LET s=bal-s: GO SUB j3: PRINT TAB 55;s$
;: LET bal=s: LET crt=crt+am: NEXT z: GO TO 3400
3250 GO SUB j1: PRINT TAB 36;s$;: LET in=in+s: LET
 int=int+s: LET s=bal: GO SUB j2: PRINT TAB 55;s$;
: NEXT z: GO TO 3400
3300 IF z$="2" THEN  GO TO 3400
3301 INPUT "Enter Slip No.";D$(X,Z,5 TO 10): IF D$
(X,Z,5 TO 10)="       " THEN  GO TO 3400
3308 LET D$(X,Z,1 TO 4)=B$
3310 ON ERR  GO TO 3310: INPUT " Enter Sale Code"'
"(press ""0"" for Help)";D
3312 FOR y=1 TO 10 STEP 2
3313 IF (D(=10)*(D)0)=1 THEN  LET D$(X,Z,11 TO 12)
=STR$ D: ON ERR  RESET : GO TO 3320
3316 INPUT ;(y);".";(I$(y)),(y+1);".";(I$(y+1));D
3317 NEXT y: GO TO 3310
3320 INPUT "Enter $ Amount";Y$: LET D$(X,Z,13 TO 2
0)=Y$
3330 IF (LEN Y$)1)*(LEN Y$(9)=0 THEN  GO TO 3320
3335 FOR y=1 TO LEN Y$
3340 IF (CODE Y$(y)(58)*(CODE Y$(y))44)=0 THEN  GO
 TO 3320
3345 NEXT y: GO TO 3215
3410 PRINT 'TAB 18;"Totals";: LET ln=24-PEEK 23689

3415 LET s=crt: GO SUB j3: PRINT TAB 27;s$;
3420 LET s=int: GO SUB j3: PRINT AT ln,36;s$;
3425 LET s=amt: GO SUB j3: PRINT AT ln,45;s$;
3430 LET s=crt-int: LET s=amt-s: GO SUB j3: PRINT
TAB 55;s$;
3440 LET D$(x,1,11)="1": IF bal(=0 THEN  LET D$(x,
1,11)="0"
3450 IF z)24 THEN  FLASH 1: PRINT "FULL PAGE, PLEA
SE  COPY  TO PAPER  AND  CLEAR WORK SHEET": FLASH
0: PAUSE o: GO TO j5
3500 IF z$="2" THEN  GO TO 2500
3950 PAUSE o: GO TO 3000
4004 GO TO 4500
4005 LET Z=1
4010 GO SUB j4
4015 IF l$="2" THEN : IF (D$(x,1,11)="0")+(a$(x,5,
14)="5"))0 THEN  RETURN
4016 IF A$(x,1,1 TO 4)="    " THEN  RETURN
4020 LPRINT ''
4021 LPRINT h$(1)
4022 LPRINT h$(2)
4023 LPRINT h$(3)
4024 LPRINT h$(4)
4025 LPRINT
4030 LET ch=93: LPRINT ''''''TAB 10;X$'TAB 10;A$(X
,1)'TAB 10;A$(X,2)'TAB 10;A$(X,3)'TAB 10;A$(X,4,1
TO 14);", ";A$(X,5,1 TO 2);".";A$(X,5,3 TO 7)''''
4035 LPRINT "Sheet:";A$(x,5,12 TO 13),"LSmt:";A$(X
,5,8 TO 9);"/";A$(X,5,10 TO 11),"Date:";B$(1 TO 2)
;"/";B$(3 TO 4);"/";B$(5 TO 6),"Terms:";T$(VAL A$(
X,5,14))
4040 FOR y=0 TO 66: LPRINT "_";: NEXT y: LPRINT
4045 LPRINT CHR$ 91;"No.* Date* Slip *Description
*Credit *Interest* Amount * Balance ";CHR$ ch
4050 REM
4102 IF A$(x,5,15 TO 21)="       " THEN  LET A$(x,
5,15)="0"
4104 LET lbl=VAL A$(x,5,15 TO 21)
4105 LET Z=1: LPRINT CHR$ 91;" 1";
4110 LPRINT TAB 5;D$(x,z,z TO 2);"/";D$(x,z,3 TO 4
);TAB 18;"Balance For.";
4120 IF VAL D$(x,z,13 TO 20)(0 THEN  GO SUB j1: LP
RINT TAB 29;S$;TAB 57;S$;CHR$ 93: LET amt=0: LET b
al=s: LET crt=ABS s: LET int=0: LET in=int: GO TO
4200
4130 LET s=VAL D$(x,z,5 TO 10): LET k=7: GO SUB j3
: LPRINT TAB 40;s$;: LET int=s: LET in=int
4134 LET k=9: GO SUB j1: LPRINT TAB 47;s$;TAB 57;s
$;CHR$ 93: LET amt=s: LET bal=s: LET crt=0
4202 FOR z=2 TO 24
4204 LET f$=STR$ (VAL b$(1 TO 2)+(VAL d$(x,z-1,1 T
O 2))VAL b$(1 TO 2))*12)
4206 IF VAL f$-VAL d$(x,z-1,1 TO 2))=VAL a$(x,5,14
) THEN  LET lbl=bal: LET a$(x,5,15 TO 21)=STR$ lbl

4210 IF D$(x,z,1)=" " THEN  GO TO 4400
4215 LET des=VAL D$(x,z,11 TO 12)
4220 LPRINT CHR$ 91;TAB 2-(z)9);z;TAB 5;D$(x,z,1 T
O 2);"/";D$(x,z,3 TO 4);TAB 11;D$(x,z,5 TO 10);TAB
 18;I$(des);
```

```
4225 IF des<8 THEN  GO SUB j1: LPRINT TAB 47;s$;: LE
T amt=amt+s: LET bal=bal+s: LET s=bal: GO SUB j2: LP
RINT TAB 57;s$;CHR$ ch: GO SUB 4960: NEXT z: GO TO 4
400
4230 IF des<10 THEN  LET am=VAL D$(x,z,13 TO 20): GO
 SUB j1: LPRINT TAB 29;s$;: LET s=in: LET k=7: GO SU
B j3: LPRINT TAB 40;s$;: LET s=(am-s)*(am-s>=0): LET
 in=(in-am)*(in-am>=0): GO SUB j2: LPRINT TAB 47;s$;
: LET s=bal-s: GO SUB j3: LPRINT TAB 57;s$;CHR$ ch:
LET bal=s: LET crt=crt+am: GO SUB 4950: NEXT z: GO T
O 4400
4250 GO SUB j1: LPRINT TAB 38;s$;: LET in=in+s: LET
int=int+s: LET s=bal: GO SUB j2: LPRINT TAB 57;s$;CH
R$ ch: GO SUB 4960: NEXT z
4410 LPRINT 'TAB 20;"Totals";
4415 LET s=crt: GO SUB j3: LPRINT TAB 29;s$;
4420 LET s=int: GO SUB j3: LPRINT TAB 38;s$;
4425 LET s=amt: GO SUB j3: LPRINT TAB 47;s$;
4430 LET s=crt-int: LET s=amt-s: GO SUB j3: LPRINT T
AB 57;s$
4432 IF lbl<0 THEN  LET lbl=0
4433 LET s=lbl: GO SUB j3: LPRINT "OVER DUE Balance
of ";s$
4434 RANDOMIZE USR 100: POKE 16090,63: LPRINT 'V$'':
 RANDOMIZE USR 100: POKE 16090,79
4436 IF VAL l$<3 THEN  LET A$(x,5,8 TO 11)=B$
4438 FOR m=(z+31) TO 64: LPRINT : NEXT m
4460 RETURN
4500 REM
4501 POKE cap,o: CLS : FLASH 1: PRINT AT o,9;"LPRINT
 FILE": FLASH o
4505 LET lbl=0
4510 PRINT AT 3,o;"OPTIONS:''''1. Print (1) statemen
t   '''2. Print Block of statements'''3. Print (1) F
ile Copy'''4.  COPY and  CLEAR Page"
4520 INPUT "Enter Option";l$: IF LEN l$<>1 THEN  GO
TO j5
4525 IF (CODE l$>48)*(CODE l$(53)<>1 THEN  GO TO 452
0
4530 PRINT '''Option Selected No.";l$
4540 IF l$="1" THEN  GO SUB 4005: GO TO 4500
4545 IF l$="2" THEN  GO TO 4600
4550 IF l$="3" THEN  LPRINT '''''''': GO SUB j4: GO S
UB 4030: GO TO 4500
4555 IF l$="4" THEN  LPRINT '''''''': GO SUB j4: GO SU
B 4030: PRINT '"New Paper Please": PAUSE o: GO SUB 4
018: FOR z=1 TO 24: LET D$(x,z)="": NEXT z: LET D$(x
,1,1 TO 4)=B$(1 TO 4): LET D$(x,1,5 TO 10)=STR$ in:
LET D$(x,1,13 TO 20)=STR$ bal: LET A$(x,5,12 TO 13)=
STR$ (VAL A$(x,5,12 TO 13)+1)
4556 LET a$(x,5,15 TO 21)=STR$ lbl
4558 LPRINT
4560 GO TO 4500
4610 INPUT "Starting Acct. No.";X$: IF LEN X$<>7 THE
N  GO TO j6
4620 GO SUB 420
4625 FOR x=1 TO 20: LET x$(6 TO 7)="0"+STR$ x: IF x>
9 THEN  LET x$(6 TO 7)=STR$ x
4630 GO SUB 4015: LET A$(x,5,15 TO 21)=STR$ lbl: NEX
T x: PRINT "Block No.";x$(1 TO 5);" Complete": PAUSE
 o: GO TO 4500
4950 LET lbl=lbl-am
4960 LET lb=2-(a$(x,5,14)="1"): LET lb=VAL B$(1 TO 2
)-lb: IF lb>=VAL D$(x,z,1 TO 2) THEN  LET lbl=bal: L
ET A$(x,5,15 TO 21)=STR$ lbl
4961 RETURN
5000 GO TO 5500
5016 LET int=0: LET bal=0: LET lbl=0: LET z=1
5019 IF (l$="1")*(n$="n")*(d$(x,1,11)="0")=1 THEN  R
ETURN
5020 IF A$(x,1,1 TO 4)="    " THEN  RETURN
5030 LET lbl=VAL A$(x,5,15 TO 21)
5120 IF VAL D$(x,z,13 TO 20)<0 THEN  LET s=VAL D$(x,
z,13 TO 20): LET amt=0: LET bal=s: LET crt=s: LET in
t=0: LET in=int: GO TO 5200
5130 LET s=VAL D$(x,z,5 TO 10): LET int=s: LET in=in
t
5134 LET s=VAL D$(x,z,13 TO 20): LET amt=s: LET bal=
s: LET crt=0
5202 FOR z=2 TO 24
5204 LET f$=STR$ (VAL b$(1 TO 2)+(VAL d$(x,z-1,1 TO
2))VAL b$(1 TO 2))*12)
5206 IF VAL f$-VAL d$(x,z-1,1 TO 2)>=VAL a$(x,5,14)
THEN  LET lbl=bal: LET a$(x,5,15 TO 21)=STR$ lbl
5210 IF D$(x,z,1)="  " THEN  GO TO 5400
5215 LET des=VAL D$(x,z,11 TO 12)
5225 IF des<8 THEN  LET s=VAL D$(x,z,13 TO 20): LET
amt=amt+s: LET bal=bal+s: LET s=bal: GO SUB 5960: NE
XT z: GO TO 5400
5230 IF des<10 THEN  LET am=VAL D$(x,z,13 TO 20): LE
T s=in: LET s=(am-s)*(am-s>=0): LET in=(in-am)*(in-a
m>=0): LET s=bal-s: LET bal=s: LET crt=crt+am: GO SU
B 5950: NEXT z: GO TO 5400
5250 GO SUB j1: LET int=int+s: GO SUB 5960: NEXT z
5405 IF l$="2" THEN  GO TO 5735
5410 LPRINT x$;" ";a$(x,1);
5415 LET s=crt: GO SUB j3: LPRINT TAB 30;s$;
5420 LET s=int: GO SUB j3: LPRINT TAB 40;s$;
5425 LET s=amt: GO SUB j3: LPRINT TAB 50;s$;
5430 LET s=crt-int: LET s=amt-s: GO SUB j3: LPRINT T
AB 60;s$;
5432 IF lbl<0 THEN  LET lbl=0
5433 LET s=lbl: GO SUB j3: LPRINT TAB 70;s$
5460 RETURN
5501 LET K=9: POKE cap,o: CLS : FLASH 1: PRINT AT o,
12;"REPORTS": FLASH o
5510 PRINT AT 3,o;"OPTIONS:''''1.  LPRINT Summery'''
2.";"Interest Calc.?"'
5520 INPUT "Enter Option";l$: IF LEN l$<>1 THEN  GO
TO j5
5525 IF (CODE l$>48)*(CODE l$(53)<>1 THEN  GO TO 552
0
5530 PRINT '''Option Selected No.";l$
5540 IF l$="1" THEN  INPUT "Print $.00 Balance Accts
.(Y/N)";n$: GO TO 5600
5550 IF l$="2" THEN  GO TO 5700
5601 LET Tint=0: LET Tbal=0: LET Tlbl=0
5605 LPRINT "Summery of Accounts","Date:";g$'': LPRI
NT TAB 2;"No.";TAB 17;"NAME";
5606 IF l$="2" THEN  LPRINT TAB 34;"Line";: GO TO 56
08
5607 LPRINT TAB 33;"CREDIT";
5608 LPRINT TAB 41;"INTEREST";TAB 53;"AMOUNT";TAB 62
;"BALANCE";TAB 71;"OVER DUE": FOR y=0 TO 79: LPRINT
"-";: NEXT y: LPRINT
5610 INPUT "Starting Acct. No.";X$: IF LEN X$<>7 THE
N  GO TO j6
5615 GO SUB 420: PRINT "Summery of Accounts: "'x$;"
TO ";
```

```
5620 INPUT "Ending Acct. No.";P$: IF LEN P$()7 THE
N  GO TO j6
5621 PRINT P$: FOR p=VAL X$(4 TO 5) TO VAL P$(4 TO
 5): LET x$(4 TO 5)="0"+STR$ p: IF p)9 THEN  LET X
$(4 TO 5)=STR$ p
5623 GO SUB 420
5625 FOR x=1 TO 20: LET x$(6 TO 7)="0"+STR$ x: IF
x)9 THEN  LET x$(6 TO 7)=STR$ x
5630 GO SUB 5015
5635 LET Tint=Tint+int: LET Tbal=Tbal+bal: LET Tlb
l=Tlbl+lbl
5640 IF INKEY$=" STOP " THEN  GO TO 5690
5650 NEXT x: NEXT p
5660 LPRINT 'TAB 17;"TOTALS";
5670 IF l$="2" THEN  GO TO 5800
5692 LET s=Tint: GO SUB j3: LPRINT TAB 40;s$;
5694 LET s=Tbal: GO SUB j3: LPRINT TAB 60;s$;
5695 LET s=Tlbl: GO SUB j3: LPRINT TAB 70;s$;
5696 LPRINT
5699 GO TO 5500
5705 LET n$="n": LET TMI=0: INPUT "Enter Annual In
terest Rate";y$: LET irt=(VAL y$/12)*.01
5720 GO TO 5600
5735 IF lbl(=0 THEN  LET lbl=0: LET int=0: LET bal
=0: RETURN
5736 IF A$(x,5,14)="5" THEN  LET lbl=0: LET int=0:
 LET bal=0: RETURN
5740 LPRINT x$;" ";a$(x,1);TAB 35;z;
5745 LET s=int: GO SUB j3: LPRINT TAB 40;s$;
5750 LET s=INT (lbl*irt): GO SUB j3: LPRINT TAB 50
;s$;: LET TMI=TMI+INT (lbl*irt)
5755 LET s=crt-int: LET s=amt-s: GO SUB j3: LPRINT
 TAB 60;s$;
5760 LET s=lbl: GO SUB j3: LPRINT TAB 70;s$
5770 LET d$(x,z,1 TO 4)=b$: LET d$(x,z,11 TO 12)="
10": LET d$(x,z,13 TO 20)=STR$ INT (lbl*irt)
5780 RETURN
5800 LET s=TMI: GO SUB j3: LPRINT TAB 50;s$
5810 LPRINT "INTEREST RATE = ";irt;" per Mth."
5899 GO TO 5500
5950 LET lbl=lbl-am
5960 LET lb=2-(a$(x,5,14)="1"): LET lb=VAL B$(1 TO
 2)-lb: IF lb)=VAL D$(x,z,1 TO 2) THEN  LET lbl=ba
l: LET A$(x,5,15 TO )=STR$ lbl
5961 RETURN
6000 CLS : FLASH 1: PRINT AT o,8;"STATEMENT HEADER
"'': FLASH o
6005 PRINT ''"1. Enter HEADER"'"2. Enter CLOSING"'
''"OPTION ?": PAUSE o
6010 IF INKEY$="2" THEN  GO TO 6500
6011 IF INKEY$="1" THEN  GO TO 6050
6012 IF INKEY$="" THEN  GO TO j5
6013 GO TO 6010
6050 PRINT "Enter Statement Header"'': INPUT "LOA
D From Tape? (y/n)";y$: IF y$="y" THEN  LOAD "head
er" DATA h$( )
6065 FOR z=1 TO 4: INPUT "Header Line ";STR$ z;h$(
z): PRINT h$(z): NEXT z
6085 POKE cap,o: INPUT " SAVE  TO Tape? (y/n)";y$:
 IF y$="y" THEN  SAVE "header" DATA h$( )
6099 PAUSE o: GO TO j5
6510 PRINT "Enter new Statment closing lines"
6520 INPUT V$
6530 PRINT '"NEW Statement Closing"': RANDOMIZE US
R 64300: PRINT CHR$ 3: PRINT V$: PRINT CHR$ 2
6540 PAUSE o: GO TO j5
7002 CLS : PRINT '"Edit File Menu:";TAB 5;"1. Chan
ge Name/Address";TAB 5;"2. Change Line Entry"
7004 IF INKEY$="1" THEN  GO TO 7500
7006 IF INKEY$="2" THEN  GO TO 7010
7008 GO TO 7004
7010 CLS : PRINT AT o,26;" Edit File "
7020 GO SUB j4
7030 GO SUB j0
7045 POKE cap,8: INPUT "Correcrt File Y/N";Y$
7048 IF Y$="Y" THEN  GO TO 7050
7049 GO TO j6
7050 ON ERR  GO TO j6: INPUT "enter Line No. to be
 changed ";z: ON ERR  RESET : IF z)24 THEN  GO TO
j6
7055 IF z(2 THEN  GO TO 502
7100 LET edit=0
7205 PRINT '''Current Line No.";
7210 IF D$(x,z,1)=" " THEN  GO TO j6
7215 LET des=VAL D$(x,z,11 TO 12)
7220 PRINT TAB 1-(z)9);z;TAB 3;D$(x,z,1 TO 2);"/";
D$(x,z,3 TO 4);TAB 9;D$(x,z,5 TO 10);TAB 16;I$(des
);: LET ln=24-PEEK 23689
7225 IF des(8 THEN  GO SUB j1: PRINT TAB 45;s$;: G
O TO 7260
7230 IF des(10 THEN  LET am=VAL D$(x,z,13 TO 20):
GO SUB j1: PRINT AT ln,27;s$;: GO TO 7260
7250 LET k=7: GO SUB j1: PRINT TAB 36;s$;
7260 IF edit=1 THEN  GO TO 7400
7265 LET edit=1
7275 PRINT '''Enter New Line No.";
7301 INPUT "Enter Date of slip ";e$: IF e$()"" THE
N : LET D$(x,z,1 TO 4)=e$
7305 INPUT "Enter Slip No.";e$: IF e$()"" THEN : L
ET D$(X,Z,5 TO 10)=e$
7310 INPUT " Enter Sale Code"''(press ""0"" for He
lp)";e$: IF e$="" THEN  GO TO 7320
7311 ON ERR  GO TO 7310: LET D=VAL e$: ON ERR  RES
ET
7312 FOR y=1 TO 10 STEP 2
7313 IF (D(=10)*(D)0)=1 THEN  LET D$(X,Z,11 TO 12)
=STR$ D: ON ERR  RESET : GO TO 7320
7316 INPUT ;(y);".";(I$(y)),(y+1);".";(I$(y+1));D
7317 NEXT y: GO TO 7310
7320 INPUT "Enter $ Amount";Y$: IF Y$="" THEN  GO
TO 7215
7325 LET D$(x,z,13 TO 20)=Y$
7330 IF (LEN Y$)1)*(LEN Y$(9)=0 THEN  GO TO 7320
7335 FOR y=1 TO LEN Y$
7340 IF (CODE Y$(y)(58)*(CODE Y$(y))44)=0 THEN  GO
 TO 7320
7345 NEXT y: GO TO 7215
7400 PAUSE o: GO TO j5
7510 CLS : PRINT AT o,10;" Edit Name "
7520 GO SUB j4
7530 GO SUB j0
7540 POKE cap,8: INPUT "Correct File Y/N ";Y$
7550 IF Y$()"Y" THEN  GO TO j6
7560 LET X=VAL X$(6 TO 7)
7625 INPUT "ACCT. NAME ";e$: IF e$()"" THEN  LET A
$(X,1)=e$
7626 PRINT A$(X,1)
7630 INPUT "St. ADDRESS";e$: IF e$()"" THEN  LET A
$(X,2)=e$
7631 PRINT A$(X,2)
7632 INPUT "RR or BOX";e$: IF e$()"" THEN  LET A$(
X,3)=e$
```

```
7632 INPUT "RR or BOX";e$: IF e$()"" THEN  LET A$(X,
3)=e$
7633 PRINT A$(X,3)
7635 INPUT "CITY";e$: IF e$()"" THEN  LET A$(X,4)=e$
7636 PRINT A$(X,4)
7640 INPUT "STATE, & ZIP";e$: IF e$()"" THEN  LET A$
(X,5,1 TO 8)=e$
7641 PRINT A$(X,5,1 TO 8)
7650 POKE cap,8: INPUT "IS THIS CORRECT Y/N ?";Y$: I
F Y$="N" THEN  CLS : PRINT "ACCT NO. ";X: POKE cap,o
: GO TO 7610
7660 INPUT "Change TERMS Y/N ? ";y$: IF y$()"Y" THEN
  GO TO 7800
7710 CLS : PRINT "TERMS MENU:": PRINT AT 2,5;"1. ";T
$(1)'TAB 5;"2. ";T$(2)'TAB 5;"3. ";T$(3)'TAB 5;"4. "
;T$(4)'TAB 5;"5. ";T$(5)
7720 INPUT A$(X,5,14): IF CODE A$(X,5,14))48 AND COD
E A$(X,5,14)(54 THEN  GO TO 7730
7725 BEEP .1,1: BEEP 1,.1: GO TO 7710
7810 GO TO j5
8000 REM close
8010 GO SUB 450
8050 CLEAR : RESTORE : RESET : GO TO 9500
9000 REM save data
9005 GO SUB 450
9010 CLS : FLASH 1: PRINT AT 0,10;" SAVE DATA ": FLA
SH 0
9015 PRINT '''Please have formated DISK Ready"
9020 PRINT '"Place SOURCE disk in DRIVE 2"''"     DES
TINATION disk in DRIVE 1"''''"Press ENTER when READY"
: PAUSE 0
9100 FOR z=1 TO 48
9110 LET c$="0100"+STR$ z: IF z)9 THEN  LET c$="010"
+STR$ z
9120 RANDOMIZE USR 100: GO TO 1
9125 RANDOMIZE USR 100: LOAD c$+"a.A$" DATA a$( )
9126 RANDOMIZE USR 100: LOAD c$+"d.A$" DATA d$( )
9130 RANDOMIZE USR 100: GO TO 0
9135 RANDOMIZE USR 100: SAVE c$+"a.A$" DATA a$( )
9136 RANDOMIZE USR 100: SAVE c$+"d.A$" DATA d$( )
9140 NEXT z
9150 GO TO j5
9500 REM
9501 DIM I$(10,11): FOR Z=1 TO 10: READ a$: LET I$(Z
)=a$: NEXT Z
9502 DATA "Merchandise","Plumb/Heat","Photo Fin.","A
ppliance","Furniture","Labor","ServiceCall","Credit"
,"Rec.on Acct","Interest"
9503 DATA "NET 30","NET 60","NET 90","NET 120","NONE
"
9504 DIM T$(5,8): FOR Z=1 TO 5: READ a$: LET T$(Z)=a
$: NEXT z
9515 LET x=20: LET CAP=23658: LET j0=300: LET j1=350
: LET j2=354: LET j3=355: LET j4=400: LET j5=500: LE
T j6=502: LET o=0
9520 DIM A$(X,5,21): DIM D$(X,24,20)
9525 DIM h$(4,80)
9530 INPUT "TODAYS DATE ? M/D/Y";B$
9535 IF LEN b$()6 THEN  GO TO 9530
9540 LET g$=b$(1 TO 2)+"/"+b$(3 TO 4)+"/"+b$(5 TO 6)
9550 LET C$="01001"
9552 DIM V$(192)
9555 LET E$=" "
9558 REM
9560 RANDOMIZE USR 100: LOAD "01001a.A$" DATA a$( )
9565 RANDOMIZE USR 100: LOAD "01001d.A$" DATA d$( )
9575 GO TO j5
9599 STOP
9600 PRINT " STOP this will erase all files": PAUSE
0
9610 DIM A$(20,5,21): DIM D$(20,24,20)
9620 FOR x=1 TO 48
9625 LET x$="0100"+STR$ x
9628 IF x)9 THEN  LET x$="010"+STR$ x
9630 RANDOMIZE USR 100: SAVE x$+"a.A$" DATA a$( )
9635 RANDOMIZE USR 100: SAVE x$+"d.A$" DATA d$( )
9640 NEXT x
9650 STOP
```

The LKDOS Auto Load Program

To Auto SAVE, type: GO TO 60.,Then touch "d".

```
   5 CLEAR VAL "63222"
  10 REM LKdos AR loader
  20 PAPER NOT PI: BORDER NOT PI: INK VAL "7": CLS
  22 LET m1=VAL "100": PRINT AT VAL "9",NOT PI;"Does
 Printer need Line Feed?    (1) Yes (2) No": PAUSE N
OT PI: LET z=NOT (CODE INKEY$-VAL "49"): RANDOMIZE U
SR m1: OPEN #VAL "3","lp": RANDOMIZE USR m1: POKE VA
L "16096",NOT PI: RANDOMIZE USR m1: POKE VAL "16092"
,(z*VAL "10"): CLS : INPUT "Enter Printer's Max Line
 length ";z: RANDOMIZE USR m1: POKE 16090,z: INPUT "
Enter Left Margin";z: RANDOMIZE USR m1: POKE VAL "16
094",(z-VAL "1" AND z)NOT PI)+(VAL "0" AND z(NOT NOT
 PI): PRINT AT VAL "8",NOT PI;"Turn Printer ON, ENTE
R": PAUSE NOT PI: CLS : LPRINT
  30 RANDOMIZE USR m1: LOAD "taswid.C1"CODE
  40 RANDOMIZE USR VAL "64300"
  50 RANDOMIZE USR m1: LOAD "AR1-9.B1"
  60 LET z=VAL "65367"- FREE +VAL "300": POKE VAL "2
3731",z/VAL "256": POKE VAL "23730",z-(INT z/VAL "25
6")*VAL "256": RANDOMIZE USR VAL "102": GO TO 5
```

NOTES:

1. When the main program is keyed in, SAVE it with a title of "AR1-9.B1".

2. This software requires a 64 column display file. You will need either the Zebra OS-64 cartridge or the TASWI code as supplied in the TASWI cassette. Either of these work with either SPDOS or LKDOS, but to use the OS-64 with LKDOS the modification given in the LKDOS manual (to stack an EPROM) is required.

****
* OTHER USES FOR THE T/S 2068 LARKEN RAMDISK *
****

LARRY CRAWFORD 357 REYNOLDS RD
LONDON ONT CANADA N6K 2P8

BACKGROUND

The LARKEN RAMDISK banks are all mapped into the DOCK bank of the 2068 even though the board is plugged into the back expansion slot. Consequently, it is possible to load any of the RAMDISK banks with data for a database or spreadsheet. This means that up to 256K more information can be handled and increases the usefulness of the "old workhorse" dramatically.
It is also possible to load a DOCK bank with code which simulates a BASIC cartridge (AROS) and to run that code as a BASIC or m/c program. This is what cartridges do.

LARKEN BANK SWITCHING

First of all, the 2068's RAM is organized into 8 chunks of 8K each. Normally, with Display File 1 in use, the bottom two contain ROM and the next one the system variables and ROM routines which are moved to RAM on power-up. These 3 chunks should not be disturbed.

The LARKEN RAMDISK board uses the top 4 chunks (4 to 7) to make up a bank. A bank is made active by switching off the HOME RAM chunks 4 to 7 and turning on one of the RAMDISK banks. The active bank is actually treated as a DOCK bank. Thus the RAMDISK board holds up to 8 DOCK banks which can be selected from the keyboard.

HOW?

Which chunks are active in HOME and which active in DOCK is controlled by the Horizontal Select Register (HSR) located in PORT 244 (F4 hex). The individual bits which make up the byte in the port determine in which bank the corresponding chunk will be active. A zero bit means the chunk is in HOME. Since we want to use the top 4 chunks in DOCK, we will load the HSR with 11110000 binary (244 decimal). Thus <OUT 244,240> activates the DOCK bank.

WHICH DOCK BANK?

LARKEN uses PORT 7 to select the one to be activated. The second page of the LARKEN RAMDISK notes gives the code numbers corresponding to the banks:

Bank	A	B	C	D	E	F	G	H
Code	7	3	5	1	6	2	4	0

Thus <OUT 7,7> will select bank A. Further, LARKEN uses bit 6 of PORT 7 to control the WRITE-PROTECT circuitry of the RAMDISK board. A zero in bit 6 puts the bank in the READ-only mode. To be able to load data into the bank, we must add 1000000 binary (64 decimal) to the bank code number. Therefore, <OUT 244,240: OUT 7,71> will permit us to put data into DOCK bank A.

WHAT CAN BE PUT INTO IT?

Strictly code. I use two of the banks to hold records for a database program. It switches the banks in or out as needed to add records or to sort them.
The code, however, can be in the form of an AROS cartridge and therefore be used to hold a BASIC program.

HOW TO SAVE THE RAMDISK BANK

Any data read into the bank would destroy the existing data. To preserve the RAMDISK, enter the following: OUT 244,240: OUT 7,7 :RANDOMIZE USR 100: SAVE "RAMDSK.C1" CODE 32768,32768. The contents of bank A will now be safely on floppy and we can mess around with the bank as much as we want. The DOCK bank, however, is still active and needs to be turned off.

HOW TO GET BACK TO NORMAL

The commands <OUT 244,0: OUT 7,0> will do the job. With a zero in the HSR via PORT 244, all bits are zeros and all chunks of RAM will be active in HOME. The zero to PORT 7 will turn return to bank H which has a code of zero. This is the normal condition for RAMDISK.

BANK H - A SPECIAL CASE

When the computer is turned on or NEWed, it checks the DOCK to see if an AROS is present. If one is, it is automatically activated. A simulated AROS loaded into this bank will run if the <NEW> key is pressed. If the power supply is turned off, the code in the bank will be lost even if the battery backup is connected. If you want a permanent AROS in bank zero, install an EPROM containing the utility in the H bank socket. I have HOT-Z in the bank and a switch in the line running from pin 1 of the 74145 to pin 20 of the H RAM chip. Without the switch, HOT-Z would run every time the computer was booted.

HOW TO RESTORE THE RAMDISK BANK

<OUT 244,240: OUT 7,71: LOAD "RAMDSK.C1"CODE> will put the RAMDISK bank A back in the RAM chip. Don't forget to <OUT 244,0>

SIMULATING AN AROS CARTRIDGE

An AROS cartridge must have 8 "overhead" bytes starting at 32768 (8000 hex) to give the computer the information it needs to run the BASIC program. See page 109 of the Tech Manual.
For our demo we will POKE into bank A the following bytes: 1,2,8,128,15,1,0,0.
The first 1 indicates that this is a BASIC "cartridge".
The 2 signals AROS as opposed to LROS, a Language cartridge.
The 8 and 128 form the beginning address of the BASIC program (low byte first). In this case the BASIC will be at 8+256*128=32776 or 8008 hex.
The 15 (00001111 bin) means that chunks 4 to 7 will be active in the DOCK bank. This is exactly opposite to the format for the HSR.
The second 1 denotes an Autostart for the program.
The two zeros reserve no bytes for m/c variables.
The BASIC program will be loaded as code starting at 32776.

CREATING AN AROS

1. Save RAMDISK bank A as described earlier and then key in OUT 244,0
   2. Key in 10 CLS: PRINT "OK": STOP
      20 OUT 244,240: OUT 7,71
      30 RESTORE: FOR X= 62768 TO 32775:
         READ Y: POKE X,Y: NEXT X
      40 DATA 1,2,8,128,15,1,0,0
      50 OUT 244,0
3. SAVE as "test.B1"
4. SAVE as "test.C1"CODE PEEK 23635+256*23636,1000
   The peeks simply find the address of the start of the BASIC program. It is moved up 50 bytes if the large printer is active. The 1000 is an arbitrary number large enough to be sure that all of the program is saved.
5. Now <GOTO 20>. This will put the "overhead bytes" for the AROS in place.
6. Key in OUT 244,240: LOAD "test.C1"CODE: OUT 244 ,0. This will put the BASIC program into the "cartridge".
7. Press <NEW> key. This activates the AROS and Line 10 of the program should run.
8. If you saved a RAMDISK bank earlier, load it back in.

RUNNING AN AROS (BASIC IN DOCK)

If DOCK bank n contains an AROS, then <OUT 7,bc: NEW> will RUN it (bc is the LARKEN code for bank n). It's that simple.

STORING AN AROS

If the code is to be left in a DOCK bank and maintained with the battery backup, it should not be in a bank that was formatted as a RAMDISK bank. If the bank is in Drive 4, then keep the AROS on floppy and load it in when you need it.

RUNNING AROS FROM DOCK AND BASIC IN HOME

Programs can exist in both banks and either one can be RUN. Which one is active depends on the contents of the aros flag at 23750. A zero at this address will allow the HOME bank program to be run. A 128 there activates the DOCK program.
A program in DOCK cannot be listed nor can it be edited. So if you attempt to enter a line and can't, you know that you are in DOCK and must return to HOME.

Steps to run both:

1. Boot the DOCK program with <OUT 7,bc: NEW>. This MUST be done first to let the computer know that an AROS is present.
2. Key in POKE 23750,0 to activate HOME bank.
3. Load in the BASIC program.
4. POKE 23750 with 0 or 128 depending on the bank wanted and then RUN or GOTO the appropriate line number.

SUMMING UP

The ability to use the RAMDISK banks for other purposes expands the usefulness of the 2068 dramatically. Data that could not be handled by the unexpanded machine can now be dealt with easily.
Huge BASIC programs can be broken down into segments, stored in DOCK banks, and called into action as needed by a short controlling program. This would leave most of HOME RAM free for the storage of data.
Expansion of the 2068 makes economic sense too, removing some of the pressure to move on to a bigger machine that can't be tinkered with.
What we need now is a mess of programs that exploit this potential.
For starters, George Chambers has two of mine. One is a database and the other a HOME/DOCK program handler. Ask him for them.

Explore and enjoy!

Larry C

****

A T/S 2Ø68 UTILITY TO PUT BASIC PROGRAMS INTO THE LARKEN RAMDISK BANKS

DONATED TO THE PUBLIC DOMAIN BY LARRY CRAWFORD
357 REYNOLDS RD LONDON ONT CANADA N6K 2P8
(519) 657-9119 27 JAN 9Ø

****

BACKGROUND

The LARKEN RAMDISK banks are all mapped into the DOCK bank of the 2Ø68 even though the board is plugged into the back expansion slot. Consequently, it is possible to load any of the RAMDISK banks with code which simulates a BASIC cartridge (an AROS) and to run that code as a BASIC program. This suite of programs will allow you to create your own simulated cartridges in the RAMDISK memory chips. A program running from DOCK leaves the HOME bank free to hold another program and/or data. Refer to Bill Jones' EXTRA MEMORY PROGRAMMING article in Update Magazine for some of the implications of working in DOCK bank.

FEATURES

See References Page 33

1. Allows a choice of any of the 8 RAMDISK banks.

2. SAVEs the chosen RAMDISK bank to floppy automatically.

3. Will RUN a program in DOCK or HOME bank.

4. Allows a BASIC program to be MERGED into HOME bank or a new program to be keyed in.

5. Will move a program from HOME bank into DOCK.

6. SAVEs the program in HOME as a BASIC program and the one in DOCK as cartridge code.

7. Reloads the saved RAMDISK bank after these SAVEs have been completed thus restoring the computer to its original condition.

8. Uses no variables internally except for the string variable Z$ so will not normally interfere with the operation of a BASIC program MERGEd with it.

9. All features are MENU driven for ease of use.

1Ø. The screen and ink colours change when you change banks so that you will know which bank is active. This information is important because you cannot LIST or EDIT lines if you are in the DOCK.

GETTING STARTED

Type in listing 1 ("maker.B1") and GO TO 9999 to save it.
GO TO 1ØØ to create and save "move2Øj.C1". This step need be done only once for each disk to be used.

Type in listing 2 ("loader.B1") and GO TO 9999 to save it. This is the program to be RUN when a new AROS is to be created or programs are to be run in both HOME and DOCK banks.

Type in listing 3 ("PRGMV.B3"). Save as "PRGMV.B3" and as a code file: <RANDOMIZE USR 1ØØ: SAVE "CORE.C1"CODE PEEK 23635+ PEEK 23636,3ØØØ>. The two PEEKs find the beginning of the BASIC program. This is necessary since LARKEN moves (prog) up 5Ø bytes if the large printer is activated. The 3ØØØ is an arbitrary number large enough to be sure that all of the program has been saved.

USING THE PROGAMS (A TUTORIAL)

1. Since the RAMDISK banks will be used, the WRITE PROTECT switch must be in the OFF position.

2. The floppy disk must have its protect sticker removed.

3. Since both the RAMDISK and floppy disks will be messed with, it is prudent to make copies of each before using the program.

4. LOAD "loader.B1". You will be asked which bank is to be used.
Enter <1> for bank "A", <2> for "B", etc.
Use <1> for a trial run.
There will be a great whirring and clanking for a few seconds while it saves the RAMDISK bank you chose, loads in the matching DOCK bank and a short m/c routine.
The screen will flicker, simulating a crash, and then some more whirring while the BASIC controller program is merged into HOME bank.
Finally, a menu in black ink on cyan paper will inform you which DOCK bank you are in and give you the choice of going to HOME bank or running a program in DOCK.
The whole process is quite scary.

5. If you choose to RUN in DOCK, the same menu will appear since the only program resident is the BASIC controller program (PRGMV.B3).

6. The other option will reveal the BASIC controller listed in white ink on blue paper.

Since the program was just MERGEd into the HOME bank, it will not autorun.
Use <GOTO 9999> to bring up the menu.

7. The menu will give you 4 options: 1: RUN, 2: MERGE, 3: MOVE to DOCK, and 4: SAVE & RESTORE.

8. Try option 1. You will be asked which bank you want to RUN in. Either choice at this time will come up with the menu for the bank chosen with the appropriate ink and paper colours.
After going to DOCK and returning to HOME, use <GOTO 9999> again to return to the menu.

9. Now try the option 2 to merge a program into HOME bank.
A CAT will be performed and you will be prompted to enter the name of the program.
The program to be merged must not include lines above 9990 or it will destroy the controller program and you will lose control.
If the program uses the TIMEX printer, it should make a USR call to 2613 to clear the printer buffer before any printing is done and should reload "mov20j.C3"CODE after using the printer.
Try "test.B1". This is a one line PRINT program at line 10.

10. Return to the menu and choose the RUN option. Choose DOCK and note that its menu comes up again because the test program has not been moved into DOCK yet.

11. Return to HOME and call up the menu.
This time choose to RUN in HOME. The test program will run and print an "OK" message.

12. Return to the menu and choose the MOVE option. A prompt will tell you to hit a key to return to menu.

13. Now RUN in DOCK and see the test program running.
Use <GOTO 9990> to get to the DOCK menu.

14. Return to HOME and delete the print line.

15. RUN in both banks and note that the test program still works in DOCK.

16. Now move the program to DOCK and RUN in both banks again.
The test program has been removed from DOCK.
(This program should be removed before using the last option of saving the DOCK program)

17. Put a line somewhere. For example, <1 REM> will do nicely.
The SAVE option deletes lines 1 to 9990 and there must be something to delete.

Choose option 4 of the HOME menu.
The program in the HOME bank will be saved as "HOMEn.B1" where n is the chosen bank.
The DOCK bank will be saved as "DOCKn.C1"CODE 32768,l. where l is the length of the file in DOCK.
The RAMDISK bank "RDBNKn.C1" will be loaded, thus restoring RAMDISK to its original state.
But the RAMDISK bank is still switched on and a program loaded in now would overwrite the data in the bank.
You will be prompted to switch off the power supply before doing anything else with the computer.

18. Now switch on and try RAMDISK. It should work normally.

****

USING YOUR AROS

Assume that you have moved a BASIC program "XX.B1" into bank 7 and want to keep it there to be run at any time.
Bank 7 cannot be formatted as a RAMDISK bank in drive 4.
The AROS was saved as "DOCK7.C1" when you created it. Change the name to "XXAR.C7" and put it into bank 7 as follows:
<OUT 244,240: OUT 7,68: RANDOMIZE USR 100: LOAD "XXAR.C7"CODE> <OUT 244,0: OUT7,0>
The program will stay in the bank as long as the battery backup is in place.
To run the AROS: <OUT 7,4: NEW>. It's that simple.
The bank can be used for other programs since the code resident in bank 7 will be saved when the new BANK7.C1 is created with the package and will be restored when the process is finished.
Thus any number of programs can be run in the one bank as long as each is given a distinctive name and is loaded into the bank as required.
File "XXAR.C7" would have to reloaded if it is desired to have it resident in the bank.

****

LISTING 1 (maker.B1)

```
   5 REM      TO CREATE THE CODEUTILITY "mov20j.C
3" WHICH SWITCHES FROM DOCK TO HOME BANK AND MOVE
S A PROGRAM FROM HOME TO DOCK,<GOTO 100>     LARR
Y CRAWFORD 20 JAN 90
  20 PRINT AT 10,4;"THIS PROGRAM WILL FORMAT";AT
11,9;"A RAMDISK BANK";AT 13,4;"WHICH BANK WILL BE
 USED?";AT 14,12;"(1 TO 8)": INPUT B: LET B$="735
16240": LET LKBNK=CODE B$(B)+16
  30 RANDOMIZE USR 100: LOAD "CORE.C1"CODE 32780:
 RESTORE 30: FOR X=32768 TO 32779: READ Y: POKE X
,Y: NEXT X: DATA 1,2,12,128,15,1,0,0,B,LKBNK,184,
11: RANDOMIZE USR 100: SAVE "DOCK"+STR$ B+".C1"CO
DE 32768,3000
```

```
 4@ STOP
 1@@ RESTORE 1@@: FOR X=23296 TO 23334: READ Y: P
OKE X,Y: NEXT X: DATA 175,@5@,198,@92,195,@4@,@14
,237,@91,@83,@92,@42,@89,@92,2@5,@69,@23,@17,@12,
128,175,211,244,126,245,@62,24@,211,244,241,@18,@
35,@19,@11,12@,177,@32,238,2@1
 11@ RANDOMIZE USR 1@@: SAVE "mov2@j.C3"CODE 2329
6,39: STOP
9999 RANDOMIZE USR 1@@: SAVE "maker.B1" LINE 1
```

LISTING 2 (loader.B1)

```
  1@ CLEAR 65367: PRINT AT 1@,@;"THIS PROGRAM WIL
L MOVE A BASIC"'" PROGRAM INTO A RAMDISK BANK"''"
  WHICH BANK WILL BE USED?"'"          (1 TO 8)":
INPUT B: LET B$="73516248": LET LKBNK=CODE B$(B)+
16
  2@ OUT 244,24@: OUT 7,LKBNK: RANDOMIZE USR 1@@:
 SAVE "RDBNK"+STR$ B+".C1"CODE 32768,32768: RANDO
MIZE USR 1@@: LOAD "DOCK"+STR$ B+".C1"CODE : OUT
244,@: NEW
9999 RANDOMIZE USR 1@@: SAVE "loader.B1" LINE 1
```

LISTING 3 (PRGMV.B3)

```
  1@ REM DO NOT DELETE THIS LINE
9991 POKE 26664,(PEEK 26664)+1: GO TO 9997+(PEEK
26664>1)+(PEEK 23750=@)
9992 CLS : PRINT AT 1@,@;"RUN PROGRAM IN"'"1: HO
ME BANK"'"2:DOCK BANK"'"CHOOSE": PAUSE @: POKE
23750,128*(INKEY$="2"): BORDER 1+(INKEY$="2")*4:
PAPER 1+(INKEY$="2")*4: INK (INKEY$="1")*7: GO TO
 @
9993 CLS : RANDOMIZE USR 1@@: CAT "",: INPUT "ENT
ER NAME OF PGM TO BE MERGED  ";Z$: RANDOMIZE USR
1@@: MERGE Z$: GO TO 9999
9994 CLS : RANDOMIZE USR 1@@: POKE 26662,(PEEK 23
641+256*PEEK 23642)-(PEEK 23635+256*PEEK 23636)+1
4: RANDOMIZE USR 233@3: PRINT AT 1@,@;"THIS PROGR
AM IS NOW IN BOTH DOCKAND HOME BANKS"'"AND YOU A
RE IN THE HOME BANK"'"PRESS ANY KEY FOR MENU": P
AUSE @: CL
S : GO TO 9999
9995 RANDOMIZE USR 1@@: SAVE "HOME"+STR$ PEEK 266
6@+".B1": DELETE 1,999@: OUT 244,24@: OUT 7,PEEK
26661: RANDOMIZE USR 1@@: SAVE "DOCK"+STR$ PEEK 2
6666@+".C1"CODE 32768,PEEK 26662+256*PEEK 26663: R
ANDOMIZE USR 1@@: LOAD "RDBNK"+STR$ PEEK 26666@+".
C1"CODE :
OUT 244,@: OUT 7,8: CLS : PRINT AT 1@,@;"HOME, DO
CK BANKS HAVE BEEN SAVED"'"RAMDISK BANK HAS BEEN
 RESTORED"'"SWITCH OFF POWER SUPPLY BEFORE"'"LO
ADING ANOTHER PROGRAM": STOP
9996 BORDER 1: PAPER 1: INK 7: CLS : RANDOMIZE US
R 23296
9997 RANDOMIZE USR 1@@: LOAD "mov2@j.C3"CODE : OU
T 244,24@: POKE 26666@,PEEK 32776: POKE 26661,PEEK
 32777: POKE 26662,PEEK 32778: POKE 26663,PEEK 32
779: OUT 244,@: RANDOMIZE USR 1@@: MERGE "PRGMV.B
3"
9998 BORDER 5: PAPER 5: INK @: CLS : PRINT AT 1@,
@;"YOU ARE NOW IN DOCK BANK ";PEEK 26666@'"YOUR C
HOICES ARE"'"1: SWITCH TO HOME BANK"'"2: RUN A
PROGRAM IN DOCK BANK"'"ENTER CHOICE ": PAUSE @:
GO TO 9996*(INKEY$="1")
9999 OUT 244,@: CLS : PRINT AT 9,7;"YOUR CHOICES
ARE"'"1: OPERATE A PGM IN EITHER BANK"'"2: MERG
E A PGM TO HOME BANK"'"3: MOVE THIS PGM TO DOCK
BANK"'"4: SAVE HOME & DOCK BANKS ";PEEK 26660;"
AND"'"       RESTORE RAMDISK BANK"'"ENTER CHOIC
E": PAUSE
@: GO TO 9991+CODE INKEY$-48
```

****

DOCUMENTATION FOR THE PROGRAMS THAT MAKE UP THE T/S 2@68 UTILITY TO PUT BASIC PROGRAMS INTO THE LARKEN RAMDISK BANKS

DONATED TO THE PUBLIC DOMAIN BY LARRY CRAWFORD
357 REYNOLDS RD LONDON ONT CANADA N6K 2P8
(519) 657-9119
27 JAN 9@

****

For background and theory refer to the tutorial Tasword file "tutor.CT" on this disk.

A RAMDISK bank is one of the RAM chips on the LARKEN RAMDISK board.
A DOCK bank is one of these chips used without being fomatted into drive 4.
An AROS is a DOCK bank file that contains "overhead bytes" and a BASIC program.

****

"maker.B1"

This short program is really two programs in one. Lines 2@ and 3@ will create a code file which, when loaded into a DOCK bank, simulates a BASIC AROS cartridge. Lines 1@@ and 11@ create the 39 byte m/c program which is called to switch from DOCK to HOME bank and to move a BASIC program from HOME to DOCK bank.

LINE 2@
Prompts for the bank to be used as an AROS. This number is assigned to variable B.
RAMDISK banks are identified as A, B, C,...,H. We will use the sequence 1, 2, 3, ...,8 instead.
This will help distinguish between the lettered RAMDISK banks and the numbered DOCK banks.
A string variable, B$, is assigned the set of integers which are the RAMDISK bank code numbers. The appropriate number is sliced out of the string, adjusted to turn off the write-protect feature, then assigned to the variable LKBNK.

LINE 3@
Loads in a previously prepared code program. More

on that later. POKES the "overhead bytes" for the AROS into the first eight addresses in the bank. Then pokes the bank number and the bank code number into the next two. The last two bytes are the number of bytes to be saved as code. The two numbers in the DATA statement represent 3000, an arbitrary number large enough to include all the data.
Then the file is saved as "DOCKb.C1", where b is the bank number.
A file of this format must be stored on the disk before the utility can be used.

.....

LINE 100
POKES the following m/c program into the beginning of the printer buffer:

```
 23296 XOR A
       LD (AROS FLAG),A  <switches to HOME bank>
       JP 3624 decimal   <the entry point to BASIC
                          in ROM>

 23303 LD DE, (prog)
       LD HL, (elin)
       CALL 5957 decimal <"difference" subroutine
                          in ROM. Loads BC with
                          difference between HL
                          and DE. Loads HL with
                          (prog)>
       LD DE,32780 dec.  <where the BASIC pgm is
                          to be moved>
  loop XOR A             <load A with zero>
       OUT (244),A       <all chunks of RAM in
                          HOME bank active>
       LD A,(HL)         <get byte of BASIC pgm>
       PUSH AF           <store byte>
       LD A,240 dec.
       OUT (244),A       <chunks 4 to 7 active in
                          DOCK bank>
       POP AF            <get byte of BASIC pgm>
       LD (DE),A         <put byte in DOCK bank>
       INC HL
       INC DE            <increase both pointers>
       DEC BC            <decrease counter>
       LD A,B
       OR C
       JR NZ,loop        <loop until counter is 0>
       RET
```

LINE 110
The 39 bytes are saved as "mov20j.C1" ready to be loaded later when needed.
Lines 100 and 110 need only be used once.

****

"loader.B1"

This is the program that is run to actually use the utility.

LINE 10
Prompts for the bank number, B, and calculates the value for RDBNK.

LINE 20
Switches on the chosen bank and saves the resident RAMDISK data.
Loads in the matching DOCK bank AROS.
Switches on all chunks of HOME bank, leaving the DOCK bank selected.

****

"PRGMV.B3" and "CORE.C1"

Both are the same program: one saved as BASIC and the other as a code file. Between them they are the controllers for loading and running BASIC programs in HOME and DOCK banks.
The code version is resident in the AROS and runs when the computer is NEWed by the loader program.

LINE 1
This is a dummy line needed if no other program is merged into HOME bank. The SAVE option deletes lines 1 to 9990. Therefore, there must be at least one line in that range.

LINE 9991
Address 26664 that is peeked, incremented, and poked, is located in an unused corner of RAM. It has a value of zero on startup and is used here as a flag to indicate whether or not this is the first run of the program.
If this is the first run, PEEK 26664 will not be greater than 1 and PEEK 23750 will not equal zero (we're in DOCK now so 23750 holds 128). Thus the GOTO will be to 9997+0+0.
If this is not the first run and we're in DOCK, the GOTO will be to 9997+1+0.
If we're in HOME bank, GOTO 9997+1+1.

LINE 9997
This line is used only on startup. First the m/c routines are loaded into the printer buffer.
The top 4 chunks of HOME bank are switched off.
Then the bank number, bank code number, and number of bytes of code in the AROS are all poked into the corner of RAM next to our flag.
These parameters must be accessible no matter which bank is in use.
Finally, the HOME bank is restored to its full size and the BASIC controller, "PRGMV.B3" is merged into place.

LINE 9998
Sets the paper and border to cyan and ink to black. These colors inform you that the computer is operating in DOCK bank.
A menu is displayed giving you the option of going to HOME bank or running a program in DOCK.

If the choice is to goto HOME then the GOTO will be to 9996*1. If not, GOTO 9996*0; thus effectively running the program in DOCK.

LINE 9996
Sets the paper and border to blue and ink to white.
These colors inform you that the computer is operating in HOME bank.
The bank-switching routine at 23296 is called and control will now be in HOME bank.
The resident program will not run automatically but must be activated with a GOTO to the appropriate line number.
GOTO 9999 will always bring up the menu in HOME.

LINE 9999
The HOME menu. First, the HOME bank is switched in and the 4 choices displayed.
PEEK 26660 in the SAVE option is the bank number.
The GOTO line is simply 9991+ the choice number.
If the first option is selected, control is passed to line 9992.

LINE 9992
A new menu is displayed giving the option of running the program in HOME or DOCK banks.
The POKE to 23750 is either 0 or 128 depending on the bank chosen. The operative bank is determined by the value stored in 23750.
Similarly, the border, paper, and ink colors are set by the Boolean logic in each of the statements.
The GOTO 0 effectively runs the program in the selected bank.

LINE 9993
Activated by the MERGE option of the menu.
It does a CAT and prompts for the name of the program to be merged.
MERGEs that program and returns to the menu.

LINE 994
Called by the MOVE option of the menu. Whatever program is resident in HOME will be placed into the AROS.
First, the number of bytes between (elin) and (prog) is found and 14 bytes added to allow for the "overhead" bytes of the AROS and our parameters. This number is double poked into 26662, replacing the previous length stored. (We had poked in 3000 originally. Remember?)
The m/c at 23303 is then called to do the actual move then a message is displayed to inform you that the move has taken place.
The menu is then called up once more.

LINE 9995
The SAVE option. Lots of activity here.
First, the program in HOME is saved under the name "HOMEb.B1" where b is the bank number.
Then lines 1 to 9990 are deleted to be sure that line 9995 is not in a chunk of RAM that will get switched off.
Next, chunks 4 to 7 are switched off for HOME bank and the write-protect turned off in DOCK.
The number of bytes to be saved is PEEKed from 26662 and 26663 and the AROS saved as "DOCKb.C1".
The RAMDISK bank that was saved at the beginning is put back in place by LOAD "RDBNKb.C1".
A message is displayed to be sure that the computer is completely back to normal by switching off the power supply before using it with another program.

****

"CORE.C1"

The controller program "PRGMV.B3" saved as code. After PRGMV.B3 is typed in, it is saved as "CORE.C1"CODE PEEK 23635+ PEEK 23636, 3000.

****

VARIABLES USED BY THE UTILITY

IN AROS	IN RAM	MEANING
32776	26660	active bank number
32777	26661	LARKEN bank code
32778,9	26662,3	length of code file in AROS
-	26664	flag for first run of pgm

****

REFERENCES

These two articles in this issue is the wrap-up on the subject of Extra Memory. There are many more articles in the back issues of Update and the total of these form a complete reference on Extra Memory, both from the Programming and the hardware standpoints. These articles will be referenced by Update Magazine issue dates below.

Oct 87-- Memory map
Jan 88-- Extra mem programming.
Mod of TS-1000 Ram.
TS-2068 whole schematic.
Ext Buss pin diagram.
Apr 88-- System Variables.
MOVE PROG - Program.
Jul 88-- DOS-64 + Bankswitching.
First of Chuck Bothner's series on RAM EXPANSION.
Oct 88-- Larken Ram Disk.
Extra memory recap.
Chuck Bothner's second art on RAM EXTENSION.
Jan 89-- LKDOS OFF Switch.
Jul 89-- FD-68 Update (bkswitch).
Oct 89-- Larry Crawford's first Ex Memory article.
Jan 90-- Two articles in this issue.

WIDJUP (Bill Pederson, 1120 Merrifield S.E., Grand Rapids, MI 49507) has some interesting facts on PRINTERS and DRIVERS that are applicable to all Computers. His CAD DISKS are for the TS-2068 LXDOS Systems. Write for information.

DOT MATRIX PRINTER DRIVER

The WIDJUP Co. graphic printer driver for 9 and 24 pin printers is a very flexible tool. It was designed to do WINDOWS. These windows refer to the printed page, not the screen.

A window can be mapped to the screen in many ways. Some can lead to confusion. If it happens to be a screen dump, it's easy to forget that it is the printer which is being mapped to the screen--not the opposite. All windows are empty until written to by programs.

Artwork programs write graphics to the screen, or to a file which can be displayed. For computers with lots of memory the files might be printer page images, often compressed, giving the program DESKTOP capability.

Most art and font programs for the TS2068 are screen oriented. That is where mapping becomes important. The printer window can be mapped to the same location that the artwork program uses. The artwork program now appears to be writing to the printer.

When the printer window is small enough to fit on the screen, it is possible to have "What you see is what you get." More properly, "You can almost see what you are going to get."

At best, you may still have distortions. WYSIWYG is a catchy phrase, but it is still an idealistic view that can't be met using any common monitor, unless dedicated for special use.

It is not wrong to want to see what your printed output will be like. That is a valid human engineering goal, though not always necessary. Good programs provide a "PREVIEW" function, often backed up with "FAT BIT" managing editors.

When a printer window is too large to be displayed across the screen, it can be divided into pieces which can. An example is displaying a 960x8 dot printer window on four consecutive screen lines with some space left unused at the end of the fourth line. This is nice for graphic mode word processing.

There are 11 pokes necessary to set up a window and map it to the screen. An application might use many windows, so some way to make this easier is required. One way is to set up an array with the most frequently used windows predefined.

The array j$(20,11) is used in WIDJUP programs. That for CAD#3 is shown below. References to a checkerboard apply to a DEMO for GRAPHONY, a utility for developing windows for applications like DESKTOP.

J$E.A$ WINDOW DATA ARRAY

	Screen Column	Corner Scan	TYPE CODE	Print Passes	Margin dots	Char Wide	LF/CR lan'	Line Feeds	Graphic MODE	Total Dots lsb	esb	
1	0	0	8	22	80	32	2	24	5	80	1	Screen Dump..Normal(9 pin)
2	0	0	68	44	40	32	2	24	5	40	1	Screen Dump..ZOOM(9 pin)
3	0	150	8	1	40	64	2	24	1	40	2	64 Wide..Lines 20-21(both)
4	0	0	8	1	0	120	2	1	1	192	3	960 dot pass #1(both)
5	0	32	8	1	0	120	2	1	1	192	3	960 dot pass #2(both)
6	0	64	8	1	0	120	2	22	1	192	3	960 dot pass #3(both)
7	0	0	8	1	40	21	1	0	5	208	0	Checkerboard top left(9 pin)
8	1	0	8	1	0	21	2	24	5	168	0	Checkerboard top right
9	0	8	8	1	40	21	1	0	5	208	0	Checkerboard white left
10	1	8	8	1	0	21	2	24	5	168	0	Checkerboard white right
11	0	64	8	1	40	21	1	0	5	208	0	Checkerboard..black left
12	1	64	8	1	0	21	2	24	5	168	0	Checkerboard..black right
13	0	168	8	1	40	21	1	0	5	208	0	Checkerboard..bottom left
14	1	168	8	1	0	21	2	24	5	168	0	Checkerboard..bottom right
15	32	32	32	32	0	32	32	32	32	32	32	$PARE
16	32	32	32	32	32	32	32	32	32	32	32	$PARE
17	32	32	32	32	32	32	32	32	32	32	32	$PARE
18	32	32	32	32	32	32	32	32	32	32	32	$PARE
19	32	32	32	32	32	32	32	32	32	32	32	$PARE
20	32	32	32	32	32	32	32	32	32	32	32	$PARE

PRINTER WINDOW PARAMETERS

Margin Dots.. These are blank dots which need not be mapped to the screen. They may be margin, column gap, etc.
Chars Wide.. There are eight dots needed per character shown on the screen, but each screen dot may be copied to the printer a selected number of times.
Total Dots.. The total of margin and multiplied char dots.
LF/CR.. Window termination may require one or both codes to allow concatenation, or cause a new line. This parameter specifies the number of codes issued.
Line Feeds.. The number of minimum line feeds to be taken at the end of the window. This can be 0.
Prt Passes.. The number of times a single window is repeated to make a larger one. This is how dumps are done.
Graph MODE.. The printer graphic mode to be used for this window. 24 pin printers do not have mode 5, but support 24 pin graphics. These added modes are not supported by this driver.

SCREEN PARAMETERS

Scrn Column.. Column 0 to 31 as usual. (Start location)
Scrn Scan.. Scan 0 to 195, from top down. (Start location)
Chars Wide.. From 1 to 120 (less for GORILLA BANANA, more for wide carriage printers). Wrap-around is active. Characters printed multiple wide reduce this.
TYPE CODE.. This byte contains width, height and video mode specifications.
BIT 6... Double wide.
BIT 5... Toggle for D_FILE2.
BIT 4... 64 wide video mode.
BIT 3... Single high. (BIT 2 = 0)
BIT 2... Double high. (BIT 3 = 0)
BITS 7,1,0... Not used.

Designers of printers cater to wants and needs of users even when those are based on less than full knowledge of how things work. The ideal of WYSIWYG has been one of these. Several of the graphics modes have been provided to achieve this for CRTs and displays of specific types, including TS2068.

The designers also provided for more enlightened users, though not always with compatibility with older methods in mind.

Dot-for-dot square printing is Mode 5 for 9 pin printers, but Mode 0 for many 24 pin printers.
Some displays emulate a printed page. 24 pin printers were the next advancement to obtain letter quality printing in many font styles that people wanted. This expanded graphics performance as well.
This new graphics capacity is only partly compatible with the older printers. As mentioned, Mode 5 was lost (though a NEARLY equivalent mode is available). 9 pin mode is lost as well.

This was acceptable because very few existing printer drivers called for these, and NEARLY was close enough to satisfy most. The impact on existing software was minimal. It is in the area of maximum utilization of graphics that there is little common ground.

Existing programs making maximum use of graphics are generally available only for other computers that use custom drivers.

CAD#2 has a driver for OLIVETTI PR2300 ink jet printer.
(1/216" line feed)
CAD#3 has a driver for EPSON 9 Pin printers and compatibles.
(1/216" line feed)
CAD#4 has a driver for STAR 9 Pin printers and compatibles.
(1/144" line feed)(Being updated)
CAD#5 has a driver for EPSON 24 Pin printers and compatibles.
(1/180" line feed)(Not available yet)

These programs differ because graphic files are unique to each printer, and that requires special routines for generation.
Data files, however, are completely transportable between them.

(C)1989,1990 The WIDJUP Co. WJP

text 87 and Quill
Part I

Peter Hale, P.O. Box 8763, Boston, MA 02114

Unless you have been on the moon for the past two years you have at least heard of **text87**, the new word processor for the Sinclair QL and Thor.

Whatever objections there are to Quill - its snail-like movement around a file and its occassional lockups - its price is right and most of us have learned to live with its 'features'.

DEFINITION: A feature is a bug that has been around so long that everyone has learned to live with it and takes pride in knowing how to avoid it.

This article, and the ones that follow, compare Quill and **text87** while at the same time provide a tutorial for using **text87**.

WHY text87?

First of all, **text87** is a WYSIWYG (What You See Is What You Get) word processor with many of the functions found in good editor programs and a lot of other goodies besides.

In truth, there may be more bells and whistles than the casual writer may want, but as a word processor it has features and is capable of work that is impossible on any other wordprocessor for any other computer.

A Brief List of the Major Goodies:

First and foremost, it doesn't ever lockup.

The cursor is faster than in Quill and erasing or moving blocks is swift like a bunny rabbit.

Virtually every printer font and enhancement can be shown on the screen in a different text color or screen font. Italic is italic, condensed is condensed, double-wide is double-wide, and it can be red, green or white. You choose. Justification and word wrap are automatic and take into account different pitches (characters per inch) of text.

The default font is proportional: an 'i' takes up less horizontal screen space than an 'm' Among other virtues, this lets you have 10% more characters to a line with no deterioration in readibility.

With right justification all spacing is micro-kerned. No double spaces to pad things out. It's almost like a type-set book, and the printer behaves the same way.

All files are in RAM. Unlike Quill, which wants to hog all available memory, it permits limits to be set on the memory to be used. Then, up to the memory available, **text87** sets a limit. However, except in its minimal configuration, it needs memory expansion.

Up to four columns of text can be printed on a single page in a single pass of the printhead. (It does not show that way on the screen - you see just one column at a time.)

A Quill _doc file can be imported to **text87** and carry with it all typefaces, tabs and margin settings, converting them to the appearance set in the program.

If you so desire, **text87** can convert the QL to a Dworak keyboard or set any key to type any character in the QL character set or any other character you choose to create. However, its up to you to mark the keys themselves.

With auxiliary files (at extra cost) screen dumps are possible and a wide range of new fonts can be designed for special effects. There is even a laser printer driver specific to **text87**.

A Brief List of Major Differences:

The first is also the first one under Goodies.

The next difference is that **text87** has no Help, as does Quill, through the F1 key. (Pressing F1 in **text87** will activate the spelling checker.)

A third difference is that the Redraw screen key is F4, not SHIFT F5 as in Quill.

Two other differences have to do with the cursor: When the cursor control moves more than one character, as with Shift Right arrow, it seeks out the position to the left of the first space

encountered; Quill seeks out the position to the right.

The other cursor difference is that it acquires its font enhancements from the character to its right; Quill picks up enhancements from the character to the left of the cursor.

The final major difference is that there is no overwrite function in **text^{87}**. The reason is fairly simple. **text^{87}** can display varying sizes of font on the screen. If the current screen font is double-wide, what happens when trying to overwrite in a condensed font?

Although disconcerting at first for a person accustomed to Quill, the new standard is not that hard to come to terms with.

USING text^{87}

If you are spoiled by the simplicity of using Quill straight out of the box, you may feel frustrated in getting up and going with **text^{87}**.

Before starting **text^{87}** for the first time, it is necessary to select the correct printer driver from the dozens of files for a wide range of dot matrix and daisy wheel printers. The one that applies to your printer must be copied to a file called driver_P87. Henceforeward, whenever **text^{87}** is loaded, this file is automatically accessed and the standards of a particular printer modify the program. (Once running you may access other driver files by name to change printers.)

Thus it is not necessary to read a printer_dat file each time a document is printed. Each printer driver includes all possible translate functions. Any character in the QL character set that is present somewhere in the printer's ROM can be used without having to run install_bas and be restricted to ten translate options.

You must write your own boot. The manual has a sample boot, but you may want to expand on it, particularly to define ALTKEYs.

After loading, which takes a bit longer than Quill, press CTRL C. **text^{87}** is EXECed, not EXEC_Wed. There are no file handling features as in Quill. This is partly to concentrate on important features of word processing, and partly because so many users now have toolkits or Taskmaster or whateever that handle files far better than Quill does and it doesn't make sense to reinvent the wheel. (**text^{87}** multi-tasks just fine under QRAM or Taskmaster.)

Some function keys are similar to those used in Quill. F2 shows the cursor status and the current Ruler. It toggles as in Quill. F3 enters the command mode and a main menu is always presented at the foot of the screen.

There are a few familiar commands: Goto, Print, Search, and Zap are ones you already know.

There is Files, but its purpose is very different. No load, save, margin, tabs, justify, footer, header, design, etc., but the functions are all there together with many others.

GOTO goes to the top or the bottom of the text, or to a page number, as well as first page, next page and top of current page, and also goes to a document line number - and boy does it get whereever it is going fast! (Using line numbers is tricky because you have to know where you are going in order to get there.)

PRINT is more complex than in Quill, but it's also more versatile. It will print a selected range of pages but also a selected range of lines. When printing text, the current cursor position is the default line number for the first line to be printed.

This command also defines headers and footers (which can be multi-lined), and the printout can be right justified even though the on-screen text is not.

Among other neat features is the option to print alternate pages. Why? You might want to print on both sides of a sheet of paper, but have the text sequential. Or perhaps page numbering needs to be at the outer edge of a page, and would require a different footer and/or header for even numbered pages than for odd numbered pages.

SEARCH has the features found both in Search and in Replace in Quill and it's a lot faster. Unlike Quill, which starts at the beginning of a file, **text^{87}** searches from the current cursor position. There's an option for Global replace-

ment, but because of its inherent dangers, it's is only offered after the first replacement. Search is case specific, meaning it ignores 'HELLO' if you ask it to search for 'Hello'. **text^{87}** remembers the prior search and/or replace string so that you may recall it and edit it to search for a modified string.

ZAP has both the document Zapping function in Quill, and the Quit, and goes through the same verification as does Quill.

FILES is different from Files in Quill. It handles load, save, merge, import or export. File names are not restricted to 8 characters as they are in Quill. They may be of any length and can include other characters not permitted in Quill including the underbar.

Saving and loading assume a _T87 extension and file names are always in capitals on the disk directory. Save is not as polite as in Quill. It does not ask permission to overwrite a file of the same name. It just does it.

Merging inserts a _T87 document at the cursor, as in Quill, but also adds any Rulers not already in the document to the table of Rulers.

There's the choice to Export in ASCII (like Printing a _lis file from Quill but making necessary conversions for Tabs) or **text^{87}** format with the extension _L87 (its primary purpose is to move text between documents.)

If a block of text has been highlit, only the block will be exported; otherwise, the whole document will be exported.

(As an aside, the rapid movement of the cursor in **text^{87}** makes block definition seem about ten million times faster than in Quill, so it gets to be very handy.)

Import offers **text^{87}** or ASCII or Quill format. The first two options insert the imported file at the cursor position. The Quill import loads a Quill _doc file as new text, and preserves all Quill highlights, converting them to the visual form used by **text^{87}**. Tabs and end of paragraph markers are retained, if desired. All import functions are very quick indeed.

CURSOR CONTROL in **text^{87}** has capabilities much as in Quill, but does a lot more. It adds the ALT key for greater versatility. SHIFT and an Up or Down cursor key move 10 lines, not just a paragraph. Add ALT key and it moves 40 lines.

SHIFT with a Left or Right cursor key sends the cursor to the start of the next space, not after it, as in Quill, but you catch on fast.

ALT plus a Left or Right cursor key takes the cursor to the start or end of the current line.

Deleting is like Quill, except that the ALT key is used with the Left and Right cursor key to remove whole lines.

As with Quill, there is a separate command to delete blocks of text.

RULERs are not found in Quill. **text^{87}** includes Margin, Tab, Justify and some Design features in one command called Ruler, but there is a learning curve to get the hang of using rulers.

The Ruler command in the main menu is used to edit and select rulers. On power up there is one default ruler (Ruler 0). Multiple rulers can be created and a selected ruler can be inserted at any point in a document. Any ruler can be copied then edited into a new ruler up to 11.5" wide.

The special feature of the Ruler Command is that its parameters are all in absolute units of length (tenths of an inch or millimetres) rather than columns, as in Quill. This means that text can be formatted according to the printer pitch chosen for printing a specific character or characters.

On screen, several pitches can be mixed without effecting the layout. Thus, if the selected printer font is double-wide, **text^{87}** knows to display only half as many characters per line as with 10 pitch pica and 10% more characters will show on a line with proportional spacing. This is what is meant by WYSIWYG.

The ruler tells the truth. Occassionally, a combination of screen fonts may force a line outside the limits of the right margin. Do not dismay. **text^{87}** knows what it is doing and what gets sent to the printer is what you want sent.

Rulers are saved with a document and it's wise to have a special file with a number of favorite rulers to load when starting a new document.

Each ruler carries a range of information:

Justification: right or left. (Actually, the choice is to justify or ignore, on the grounds that justify means right justification.) There's no Center Justification; use Center Tab instead.

Line spacing: the space between lines can be adjusted to fractions of an inch. The default is 1/6 of an inch (12/72 or 30/180, depending on the printer) but can be modified to your heart's content. Thus it is possible to have double spaced text mixed with single spaced text, depending on the selected ruler.

Margins: left, right and indent. Indent defaults to left margin, but may be designated as to the right where the first left tab is located if it is automated from a special indent feature in the Mode command. The Indent mark itself is exclusively for setting Indent to the left of the left margin to create a hanging paragraph.

Selection: only one ruler at a time is active under the cursor. A ruler is selected as an option from the Ruler command. (The current ruler is shown when the F2 key is toggled on.) The selected ruler is inserted at the cursor when SHIFT F5 is pressed.

Tab stops: left, right, center and decimal.

Enough already!

The next article in this series will explore block handling and the numerous features in **text^{87}** that make screen presentation truely revolutionary.

This article has been prepared on **text^{87}** mostly in 15 pitch type. Note the use of micro-kerning.

You may order **text^{87}**, version 3.01, from Software 87, 33 Savernake Rd., London, NW3 2JU United Kingdom for £60. It includes Qtyp as an integrated spelling checker with English, French and German dictionaries. It is also available in the USA for $94.95 from EMSoft, P.O. Box 8763, Boston, MA 02114-8763 (617) 889-0830.

EMSoft

Software for the QL that works hard

Now has in-stock the very best true WYSIWYG QL word-processor: text^{87} v3.00 with spell check and several new features. (includes 1 hr USA help line). $94.95

Tax-time means	TAX-I-QL/89	$24.95
RECIPE	_dbf	$14.95
INVENTORY	_dbf	$9.95
TRUST_FUND	_dbf	$24.95
QLAND_LORD	_dbf	$24.95
MAILBAG	_dbf	$24.95
DBTutor	_dbf	$24.95
DBProgs	_dbf	$19.95
CDIR/BOOTS	SBas	$9.95

On 5.25" DD or QD disks. On 3.5" or microcartridge add $2.00.

FREE CATALOGUE

EMSoft
P.O. Box 8763
Boston MA 02114
(617)889-0830

Two tough things to find for your Timex or Sinclair ZX/TS computer--

SOLID INFORMATION & GOOD CONNECTORS

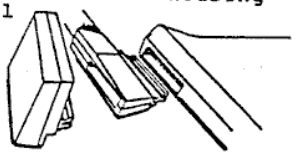
YOU KNOW WHERE TO GET THE BEST INFO FOR YOUR ZX/TS--SYNTAX, SQ, & THE FOYT BOOK CHECK IT OUT--ASK ANYONE-- THEN GET 'EM ALL FOR

JUST $19.95 plus $10 S&H

BUS CONNECTOR & EXTENDER & HOUSING

LIKE A PRINTER CONNECTOR

Keyed Connector, Extender PC Board
PLUS Molded Black Plastic Housing
Fits: ZX80 & 81
TS1000 & 1500
PC8300
ZX/TS SUBSET
OF SPECTRUM
& 2068



Pkg of 3, ONLY $13.95 plus $2 S&H
SAVE Buy 2, get a FREE one
9 for $26.95 + $6 S&H SAVE

EMS, Box 8763, Boston, MA 02114-8763
617-889-0830 VISA & MC Add 4%

TS-2068 Programming In the Abstract

THE STACK: The manipulation of the STACK comes under the heading of "where only fools dare tread". But aren't we brave? Look at it this way, "you cannot hurt anything by trying, so why not?". AND, we might be enlightened. Key in the following little program and SAVE it. Then we'll traipsie through the stack a bit.

```
10 FOR N=25088 TO 24833 STEP -1
30 PRINT n;"="; PEEK n;: NEXT n
60 PRINT "RETURN Line="; PEEK 25083+256*PEEK 25084
'"Statement #="; PEEK 25085
70 STOP: GO SUB 90
90 STOP
```

OK. First off, we will look at what we can and cannot do with the Stack in BASIC. Timex intended the stack to be "hidden from the user". But a small tydbyte of information is given in the user manual, page 255. Take a minute to look at the sketch in figure C-1. The little program given above PEEKS this area. The Stack is the TS-2068's exclusive "library" of "pending things to do". If <GO SUB> is used in a program line, the LINE NUMBER is factored into a 16 bit word and that word is stored on the stack at address 25083 and 25084. Then the "next statement number" in the line after the GO SUB is stored in address 25085.

In our little program, the GO SUB is in line #70, statement 2. So, if allowed to execute, the number 70 would be stored in address 25083. Zero would be stored in 25084, and 3 would be put in adr 25085. Check this out by RUNNING the program. As the program runs the contents of the Stack addresses will be printed to screen. Somewhere at around address 25000, a train of zeroes will print. These Zeroes tell us the "FREE MEMORY" spaces of the stack. As more and more "library" items are placed on the stack there will be few Zeroes in the train. These "TS-2068 Library items" are put on the stack to be used, and once used they are erased. All of this is done by the internal management utilities in the TS-2068 ROM. BUT, we can have a little influence on things.

When the program runs the first time, a report will be given. This "RETURN LINE # and Statement Nbr." will be inaccurate. This is because there is no RETURN placed on the stack at address 25083, 25084, and 25085. So, ignore the report on the first run. The program will STOP at line #70. When it does, type CONT ENTER. Then another STOP will occur at line 90. This time DONT do anything for now. Now lets look at what we did.

We have introduced a "piece of JUNK" on the stack, because we did not use <RETURN> after the <GO SUB>. But in this case we do it intentionally in order to see the RETURN in the Stack. After the STOP at line #90, type GO TO 10 ENTER. Write down the numbers given for address 25083, 25084, and 25085, during the first screen full of numbers. Then let the numbers scroll to the end. At the end report the line #70 will be reported, with statement number 3. So, we have captured the <GO SUB that is on the stack>. AND the STOP again occurs at line #70.

Now lets work a bit with 16 bit numbers. First add LINE # 2000 to the program. add: 2000 STOP. What is the 16 bit number for 2000? ans: 2000/256=7.8125. Drop the decimal and 7 is the msb (most significant byte). The lsb (less significant byte) is gotten by: 2000-7*256=208. So, the 16 bit number of 2000 is "208,7". Now we will CHANGE the line number for the RETURN that is on the stack. POKE 25083,208:POKE 25084,7:POKE 25085,0. After these POKES, Type <RETURN ENTER>. Note the STOP at line #2000. Now we have learned how to change a <RETURN> on the stack to a different line number in a basic program, but WHY would we ever want to do that?

Well I'll give you a small example and you will think of others. Suppose that after a menu we want to present on screen the data contained in A$. But if A$ has no data, we want the program to RETURN to a different program line than contains the GO SUB. I'll construct the routine:

```
500 (MENU item 4 selected)
510 IF Z=4 THEN GO SUB 900
900 IF LEN A$<1 THEN POKE 25083,208: POKE 25084,7:
RETURN
910 PRINT A$: RETURN
2000 REM Routine to use if A$ has no data.
```

RULES: You must have a GO SUB on the stack before POKING the addresses 25083, 25084, 25085. Otherwise you CRASH. So, the POKE must be done AFTER the GO SUB. Dont poke other Stack Addresses, or you'll get a crash. Oh well, what the heck, and why not?

JUNKY STACK: If you want to find out how fast the stack can be completely corrupted, Remove the Stop at line 70 and change line #90 to read: 90 GO TO 10. Then RUN. One of the pit-falls of using <GO SUB> is that BASIC programs can be broken into. Each time a GO SUB routine is broken into three addresses in the Stack are corrupted.

USING BASIC STRING / ARRAYS IN MACHINE CODE PROGRAMS

Dear Bill,

by Charles Bothner

I looked over your letter concerning the use of BASIC variables and arrays inside machine code programs.

You said that you needed to locate a specific variable in the BASIC VARS area of memory in order to use the data in a machine code program. The following procedure will allow the transfer of the location and size of a simple variable to a BASIC program. If you want to locate a variable which has not been fixed in size by a DIM statement, insert the following lines at appropriate line numbers in your program:

```
1000 POKE 23728, CODE"W"
1010 LET X=USR 23296: REM This is the start of printer buffer
1020 LET SIZE=PEEK (X+1)+256*PEEK (X+2)
1030 LET START=X+3
```

Line 1000 puts the letter code for the variable into a convenient storage location, in this case into an unused slot in the normal system variables table. Line 1010 accesses the machine code to locate the variable address in memory. In this case I have put the 28 byte routine into the printer buffer, but any convenient location could be used since the code is completely relocatable. On return from the MC routine, the variable X contains the address of the identifier for the W$ variable. Line 1020 reads out the current length of the W$ variable and line 1030 gets the starting address of the actual characters in the variable. If desired, these values may be POKEd into a convenient memory location for subsequent access by whatever MC routine you intend to use to manipulate the data. Here is the machine code which locates the starting address of the W$ (or any other) simple variable:

ADDRESS		OPCODE	INSTRUCTION	COMMENTS
5B00		3AB05C	LD A,(5CB0)	Get character code
5B03		CBAF	RES 5,A	Insure upper case char.
5B05		2A4B5C	LD HL,(5C4B)	Get VARS area address
5B08	LOOP:	BE	CP A,(HL)	Test variable identity
5B09		280E	JR Z, FOUND	Jump if right variable
5B0B		FE80	CP A, 80	Is this end of VARS ?
5B0D		2002	JR NZ, CONT	Continue if not end
5B0F		CF	RST 08	Error return otherwise
5B10		01	VARIABLE NOT FOUND message ident.	
5B11	CONT:	23	INC HL	Point to size of field
5B12		5E	LD E,(HL)	Get size of variable
5B13		23	INC HL	field size into DE
5B14		56	LD D,(HL)	
5B15		23	INC HL	Point to next location
5B16		19	ADD HL,DE	Point to next variable
5B17		18EF	JR, LOOP	Test this variable
5B19	FOUND:	E5	PUSH HL	Get the address into BC
5B1A		C1	POP BC	
5B1B		C9	RET	Done

This routine scans the entire VARS area of memory until the starting address of the specific variable storage area is located or until the end of the VARS area is reached, at which time there will be an error type return. This will happen if the variable W$ (or any other simple string variable which is being searched for) has not been previously assigned. If the variable has been located, then the address is returned in the BC register pair. Line 1010 in the BASIC code above will assign this value to the variable X on the return from the machine code.

If a search for a fixed dimension string array is desired, it will be necessary to insert the following assembly code bytes between the addresses 5B04 and 5B05 in the routine listed above:

```
C680        ADD A,80H     Converts to proper type
                          of dimensioned array
                          identifier
```

It will be necessary to modify the lines of BASIC code to determine the actual size of the stored character field in the array due to the extra data bytes stored in the data field for this type of variable (see page 258 of the TIMEX 2068 users manual for the details). Since the array has been previously dimensioned in the program [e.g. DIM W$(40,256)], we can determine the correct offset by taking the number of array dimensions (2 in this case) and doubling it, then adding 1 for the number of dimensions byte in the variable's data field and subtracting the result from the data obtained in line 1020 above.

For the example of W$(40,256) the changes to the BASIC lines is as follows:

```
1020 LET SIZE= PEEK(X+1)+256* PEEK(X+2)-(1+2* PEEK(X+3))
1030 LET START=X+3+(1+2* PEEK(X+3))
```

Obviously the terms can be combined to shorten the lines but I left all the terms separate to clarify (I hope) the procedure to locate the actual starting address for the character data stored in the array along with the actual length. The size of the actual character data field could be determined in this case by simply multiplying the terms in the DIM statement. It is better to determine the size on the fly for the cases where the array dimensions may be determined by another variable instead of a fixed value [e.g. DIM(256,Z)].

I hope that this brief explanation will give you an idea of how to operate on data stored in the VARS area by either BASIC or machine code routines. To summarize, the transfer of data from the BASIC program to the machine code requires the POKEing of the data into memory locations where it can be accessed by the machine code. Transferring of a single character or numeric (integer) value from the machine code to the BASIC program may be accomplished by loading the data into the BC register pair prior to the return from the machine code.

If this information is not what you really wanted or if additional clarification is needed, please let me know.

best regards,

Chuck B.

Chuck Bothner

Al Feng is treasurer of the Chicago TSUG (CATSUG) and an expert programmer of the Sinclair QL. Al takes time out from his writing for "Nightime News", the CATSUG Newsletter, to give us information on three interesting subjects.

QDOS ROM Code

By Al Feng

Most Sinclair QL users in North America are already aware that the Operating System (OS) in their computer is called QDOS (i.e., QL Disk Operating System). This may seem as a bit of a misnomer since the QL comes with microdrives for storage; and, all disk drive interfaces are third party! In fact, QDOS allows (i.e., "forces") you to select the device being addressed by the user (e.g., 'mdv1_', 'flp1_', 'ram1_', 'win1_'). Clearly, either Sinclair Research Ltd. or Tony Tebby realized that Uncle Clive's microdrives may NOT be the ultimate storage device.

For those who are not overly familiar with the inner workings of their QLs, the difference between "British" and "North American" QLs involves more than BT (British Telephone) connectors vs. "D" connectors. In addition to evolution in the PCB (Printed Circuit Board) layout (i.e., the hardware), various versions of the QDOS were released (i.e., the firmware).

Not knowing which QDOS version you are running is generally not a handicap or inconvience. For practical purposes, a QL with an 'AH' ROM will run the QUILL word processor as well as one with an 'MG' ROM. The transparent difference is predominantly in the ERROR handling (i.e., the user usually doesn't see it).

To find out what version QDOS your QL is running (all QLs made by SAMSUNG ELECTRONICS COMPANY Ltd. in Korea have 'JSU' ROMs) type:

```
PRINT VER$          <ENTER>
```

There are only a couple of reasons to know what QDOS ROM code your QL has. The primary one is to satisfy your curiousity. The second is if you are running some programs (e.g., compilers which depend on EXTENSIONS to QDOS). Often times, a "patch" is included to compensate for incompatibilities in the earlier code, so the versions still doesn't matter!

Okay, what do those designations mean? Does the 'AH' designation mean that they think they finally got it to a satisfactory point ["AHhhh"]? Does the 'MG' designation refer to the Morris Garage vehicles? Who can say for sure except the person/people who chose the designations! After all, the "GW" in 'GW-BASIC' stands for "GEE WHIZ" (No kidding!).

A half dozen years ago, I asked my brother (an IBM PC user) why PC-DOS (the IBM "specific" version of MS-DOS) had to be LOADed via software rather than being a permanent part of the system (that is, as firmware on a ROM or EPROM). In fact, I asked a lot of "almost stupid" questions back then. To make a long story short, he replied that the Disk Operating System (DOS) was loaded as software so they (Microsoft, in this case) could implement any subsequent changes to the code more easily.

At the time, the idea of changing the code didn't makes sense. After all, why re-write code if it was written correctly the first time! Okay, my computer experience was limited at the time, and I was a bit naive about programming; but, I know someone who thinks that the ABACUS Spread Sheet programmers should have been able to write the program in about 10K (yes, 'ten')!

Among the "almost stupid" questions I asked was with regard to the differences (if any) between DOS 2.0 and DOS 2.1. But, I digress. Very simply, a change in the integer represents a "major" change to the code, while "minor" changes are represented by variances in the decimal or hundredths place. In theory, higher numbers indicate improved code (as with ERROR trapping). Latter versions generally have "improved" code (i.e., more ERROR trapping).

If you will make the allowance for a comparison between the exceedingly crude DOS 1.0 for the INTEL 8088/8086 chip series [e.g., CLS (CLear Screen) was not included] and the very earliest QLs which had a "dongle" (as the Brits apparently refer to it) on the backside of the QL then from my reckoning and recollection there are rough analogues which one can make between MS-DOS (Microsoft DOS) and QDOS.

Sinclair Research Ltd. (or, Tony Tebby) chose to use an alphabetic system for labelling. Just as there were various MS-DOS versions, so there are various QDOS versions. For example, we may compare:

--[QDOS]--	--[MS-DOS / PC-DOS]--
AH	2.1
JS	3.30
MGUK	4.00

QDOS is resident in the QL when the computer is powered on because the code is contained on two socketed ROM chips. Several years ago, Tom Bent designed a small PCB which utilized a 27C512 EPROM which could substitute for the two ROM chips (see QUANTUM LEVELS 1:6 for schematic). A big plus to Tom Bent's design is that it lets your QL run cooler.

The 'J' series is probably the most predominant. The 'JM' & 'JS' are the two most prevalent versions still being sold in the UK. The 'JSU' ROM is the same as the 'JS' ROM with the exception of the size of the video scan when in TV mode. The 'JSU' ROM accomodates the lower resolution of NTSC televisions than the British PAL televisions generate (512x212 vs. 512x256).

Since the TV-mode on all the "British" ROMs generate a screen having the same dimension as the monitor display, the apparent difference is that initializing with 'F2' should allow generation of eight colors rather than four [unfortunately, I forgot to verify this when I

had temporary access to an RGB monitor].

Programs can "sense" which mode the computer is in by PEEKing address 163890. In monitor-mode, the value returned will be '0'. On all "British" ROMs in TV-mode the value returned should be '1'. On the 'JSU' ROM and the "rogue" 'MGSJ' ROM, PEEKing this address when in TV-mode should return a value of '2'.

Just as there are improvements to Microsoft's DOS version 4.nn, so too are there improvements to the 'M' series ROM. Unfortunately, the price of improvement may be incompatibility with code which is LBYTEd/LRESPRed. The conflict / compatibility varies with the code. For example, early (all?) versions of 4MATTER will not run on a QL having an 'MGUK' ROM. Code which is "automatically" initialized on a 'J' series ROM (such as SPEEDSCREEN RAM code) must be further beckoned with the 'M' series ROM code.

It has been reported that Digital Precision's IBM emulator would not run on a QL having 'MG' code. Details are not complete; and, I suspect he may have the 'MGSJ' code as I have personally seen THE SOLUTION running on an 'MGUK' ROM. *(Ed note: My MGUK ROM QL (installed by Tom Bent) runs both Solutions and PC Conqueror)* In theory, the 'MGSJ' code incorporates the NTSC television screen scan with the rest of the 'MG' code. This may or may not be a consideration.

Those of you familiar with MS-DOS will find a comparison between the 'MGUK' ROM and DOS 4.00 appropriate. It seems that enough nuisances were unintentionally written into the 'MG' code that it probably makes the more familiar 'J' series the code of choice (my opinion).

Just as "other" DOS exists for IBM compatible computers (e.g., WENDLIN-DOS), there are "proven" alternatives to QDOS. The first of these is 'ARGOS' code which is used by CST (defunct Cambridge Systems Technology) on their THOR computers [DANSOFT is marketing the THOR, but reports suggest that they are NOT shipping]. The second is 'MINERVA' by QVIEW.

'ARGOS' is apparently QDOS which has been tweaked to support the THOR's hardware. It may NOT be hardware compatible with the QL.
Unlike 'ARGOS', 'MINERVA' code was apparently reverse-engineered to accomplish the same functions as the QDOS code. The resulting code is supposed to be superior in speed, particularly in the screen handling routines. As of mid-March, 'MINERVA' is only available directly from QVIEW (in England).

If you're thinking of replacing your QDOS ROM code, I recommend that you consider waiting for 'MINERVA' code to become available. If you can't wait, then opt for the 'JS' code; after all, 'JSU' code is available on your "spare" QL when don't have a monitor.

HAPPY TRAILS, AND COMPUTING, TO YOU! Al Feng.

Ed Notes: Al touched upon the main difference in design of QDOS and MS-DOS, which is: QDOS is permanent in ROM while each MS-DOS utility must be "loaded from disk to RAM" to operate. There are advantages to both methods. Al mentioned the advantage of MS-DOS being that it can be changed easily by recording another DOS version on floppy disk. The advantage of QDOS being in ROM is that its functions are "already in memory" and thus is faster (given the same clock speed and efficiency of operation). I have often wondered why some smart designer hasn't made use of "EEPROM" to store a DOS. ELECTICALLY ERASABLE PROMs have the ability to be written to by the computer (about 3 million times for each memory address). That should be sufficient for as many DOS version re-writes as needed, wouldn't you think?

Ed Grey Enterprises, PO Box 2188,

Englewood, CA 90305 was to send in an advertisement of his wares. I'm sure that Ed will kick himself in the britches when he realizes that he missed the boat. Instead of using one of his old ads, I'll just make some comments. Ed Grey carries a whole line of TS-2068 support items and runs a West Coast USA BBS called *"The Grey Matter"*. His Telephone number for ordering software and hardware support items is 213-759-7406. The BBS telephone is: (213-971-6260). This BBS is one of the most informative and the most used by Sinclair Timex users. If you want to talk to an enthusiastic Sinclair Timex person, call Ed. If you want to find out what goes on in Sinclair Timex Computing, Call the BBS (with your computer).

This UPDATE Issue:

And Others Ahead:

The problem with this issue was the DECISIONS about what to include! The articles were selected from about two issues worth of excellent material given freely by your fellow Sinclair Timex users. Your thanks should go to the authors, not to yours truly.

There several other excellent articles about the QL and TS-2068 that are awaiting space in the magazine. They aren't "culls", but are of a length that just couldn't be included in this issue.

FRANK DAVIS, your new publisher will need this same outstanding support from the writers in order to continue the depth of coverage that has been a tradition with Update Magazine. I feel sure that all of our outstanding writers will continue to support the users through the pages of Update.

RE-SUBSCRIBE NOW! You have ONE more issue of your subscription coming (July). Frank Davis, the new publisher, needs to build his 1991 subscriber list and needs to have as many early subscribers as possible for his planning. *Please send your $18.00 subscription renewal to:*

FRANK DAVIS, PO BOX 1095, PERU, IN 46970.

A PRIMER FOR USING A QL-MUTILATOR

by Al Feng

Okay. Just kidding. We'll discuss MS-DOS with the QL instead. But, after hearing "all" those bad things about MS-DOS (Microsoft's Disk Operating System for IBM compatible computers) one can only presume that running a SINCLAIR QL in emulation mode has to be a mutilation of the QDOS operating system. One would like to think that it wouldn't happen; but, the price of software emulation code is still less than an MS-DOS compatible computer.

The following isn't going to be a discourse about whether you should be using an emulator on your QL. Only you know whether there is a need "at home" for MS-DOS compatibility; and, if an emulator is going to be a satisfactory route.

The following is going to tell you how to overcome some of the conceptual inconveniences of MS-DOS whether on a QL, or on an MS-DOS specific computer.

The first thing you have to know is that drives are described alphabetically rather than numerically. Thus, a drive may be labelled 'A' through 'Z'. The Operating System (OS) knows you are describing a drive because a colon (:) is used as a delimiter immediately after the letter. The reason for this is because a file may be named 'A.BAT', 'A.COM' or 'A.EXE' and be otherwise executed if you simply INPUT 'A' followed by the '<ENTER>' key. QDOS uses an underscore ('_') as the general delimiter. Thus:

```
A:  ==  flp1_
B:  ==  flp2_
```

The major difference is that QDOS allows/forces you to describe the physical/virtual device being addressed. On the other hand, MS-DOS assigns labels on a "logical" basis unless otherwise specified (some rules do apply).

On an MS-DOS system, if you have two floppy drives, they are 'A:' and 'B:'. If you have a hard drive, it is 'C:'. If you have one floppy drive and a hard drive, the latter is still 'C:' (I told you, some rules apply). If you have two floppies, no hard drive, and have set up a ram drive, then the ram drive becomes 'C:'. If you have two floppies, a hard drive, and a ram drive, then the ram drive becomes 'D:'.

Ram drives are static and are configured during BOOT-up via the CONFIG.SYS.

Even if you have a limited amount of overall ram in your system, but more than 384K, then you should seriously consider setting up a small ram disk on your mutiliated QL, laptop, or early PC. A 48K drive is sufficient for DOS 3.nn and 72K for DOS 4.nn.

An example of a CONFIG.SYS for DOS 3.nn where the size of the ramdrive is indicated by the number following the "device=" statement (in this case '48K'):

```
buffers=15
files=30
device=ansi.sys
device=vdisk.sys 48
```

Unfortunately, the boys-and-girls at Microsoft are sloppy coders (Captain Midnight would not be happy), and the same files in DOS 4.nn tend to be larger than in DOS 3.nn. For DOS 4.nn your CONFIG.SYS file will be similar to the following (note the name change from VDISK to RAMDRIVE):

```
buffers=15
files=30
device=ramdrive.sys 72
```

Okay, 48K is an awfully small ram disk. What are we going to put in it (and, why?)? There are two files which should be transferred to a ramdrive regardless of the size and speed of the system being used (this includes AT classs machines). The first is COMMAND.COM and the other is FORMAT.COM.

As these files were designed, they are what are referred to as external DOS commands. An external command is not retained in the systems memory. This is because DOS was originally designed to run on a machine with as little as 16K of memory.

The ability to initiate a program via COMMAND.COM and/or format a disk via FORMAT.COM were considered task oriented. After a program is LOADed, DOS dumps the COMMAND.COM from memory, and brings it back when the program is quit. This is why you may see a message which instructs you to "replace disk with COMMAND.COM in drive A:". This can be circumvented by using the statement:

```
set comspec=c:command.com
```

where 'c:' is the location where the COMMAND.COM can be found (in this case a ramdrive ... adjust accordingly). This can be indicated in a line in the AUTOEXEC.BAT.

While COMMAND.COM is always necessary, FORMAT.COM is not. It is, however infrequent a task, especially inconvenient (i.e., "slow") to have FORMAT.COM load from a disk rather being resident. By transferring an appropriate FORMAT.COM to a ramdrive, you have turned it into an almost-internal command
(using FORMAT.COM will be discussed later). An example of a true internal DOS command is COPY.

An example of your AUTOEXEC.BAT should be similar to one of the two following examples:

```
@ECHO OFF
PATH A:
PROMPT $P$G
copy format.com c:
copy command.com c:
set comspec=c:command.com
copy dr.com c:
copy free.com c:
c:
```

OR,

```
@ECHO OFF
PATH A:
PROMPT $P$G
copy format.com c:
copy command.com c:
set comspec=c:command.com
copy dr.com c:
c:
```

If you need MODE.COM, then include it as a line of your AUTOEXEC file.

FREE.COM is from the first PC MAGAZINE Utility disk. DR.COM was a freebie from PC/COMPUTING magazine.

FREE.COM tell you how much "free" space is left on a disk. This sounds, well, stupid. But, if you have blank, FORMATted disks without files, you cannot determine the size with the DIR command. In these instances, FREE.COM will let you determine if it is a 360K or 720K disk:

```
C:> FREE B:
```

DR.COM (actually DIRMAGIC.COM) is a convenient front end program which was distributed "free" if you subscribed to PC/COMPUTING. It allows you to COPY/DELETE/RENAME/VIEW/etc. from a dynamic menu.

FREE.COM and DR.COM are freeware (vs. shareware) programs which are supposed to be availble on various bulletin boards. You can probably get a copy of these two files from Bill if you send him a few of dollars (?) to cover media/packaging/postage and tell him which disk format (720K or 360K) you need.

The reason the example of the AUTOEXEC.BAT above ends with a 'C:' statement is that a lot of activity can be initiated from the 'C:' drive:

```
C:> COPY A:*.* B:
```

is the same as: WCOPY flp1_ to flp2_

```
Compare:    C:> COPY A:DR.COM B:

            A:> COPY DR.COM B:

            B:> COPY A:DR.COM
```

to: COPY flp1_DR_COM TO flp2_DR_COM

Also compare:

```
C:> COPY A:DR.COM B:DIRMAGIC.COM
```

to: COPY flp1_DR_COM TO flp2_DIRMAGIC_COM

The syntax of a DOS statement is clearly more compact since the current "path" can be implicit in the statement. Working from a "C:>" prompt is similar enough to working with a command cursor in "WINDOW#0" that you can consider performing general file management from this drive.

Similarly, if you want to delete a file you would use the DEL statement:

```
C:> DEL A:DR.COM

A:> DEL DR.COM

B:> DEL A:DR.COM
```

to: DELETE flp1_DR_COM

If you are going to LOAD an application (program), then INPUT the drive designation followed by '<ENTER>'; and, then the file name. For example:

```
C:> A:          <ENTER>

A:> QUILL
```

The syntax for FORMAT.COM with an emulator is similar to the FREE.COM utility:

```
C:> FORMAT B:

and,        B:> C:FORMAT

OR:         C:> FORMAT A:

and,        A:> C:FORMAT
```

With the emulating QL the density is determined by the emulating code. There are conditions which may be needed if you wish to FORMAT to a density other than the drive's normal specification.

UH-oh. Was mention made of how to construct your AUTOEXEC.BAT and CONFIG.SYS files? Yikes. The MS- [Masochistic & Sadistic] method is for you to use EDLIN! If you can get a copy of DR.DOS's (Digital Research's DOS) EDITOR.EXE then you will have an easy-to-use line editor. If not then use ANY word processor:

Set the left margin to zero ('0');
the upper and lower margins to zero; and,
ensure that there is no header or footer.

Next, transcribe the appropriate example(s) as individual lines of text. SAVE each individually as an appropriately named ASCII file. Transfer the file to your DOS disk.

That's about all you really need know to be functional on the precarious, MS-DOS iceberg.

HAPPY TRAILS, AND COMPUTING, TO YOU. Al Feng

THE QUANTA LIBRARIAN FOR NORTH AMERICA?

Will the active QUANTA LIBRARIAN please get in touch with Update Magazine. Why? Well, I'd like to help publicise this fantastic group of programs and utilities. I have had the first 15 Quanta Library Disks for about a year. There hasn't been much published about the utility of this good library or how the users can get the disks. Perhaps there are enough interested subscribers to form a Quanta Group in North America? Update will help.

THE CANADIAN REPORT

Kanadian fonetik langwaje

TORONTO, Canada (KNT) — Like a solitary savior striving to rescue the illiterate from the flummadiddle and kerfuffle of the English language, Ted Culp envisions "that gloreus day all linggwistic owtrajes wil end!"

At that inevitable moment when English collapses under the weight of its confusing, irrational structure, Culp says he'll be ready with the Canadian "langwaje," a sane replacement for the "garbage" that annoys people and produces illiteracy.

"As this inferior langwaje iz displased everywhare ... illiterasi shud disapeer completely," wrote Culp, the meticulous publisher of the *Times Ov Toronto*, and a teacher and ardent proponent of simplifying a language that drives him nuts.

In a multiethnic country where combat over language could qualify as a national pastime, many view a phonetic-type spelling of English as laughable or an uncomfortable reminder that some really spell like this.

Others view Culp's changes as an assault on the Queen's English. After all, many loyal Canadians still sing *God Save the Queen.*

But this is no joke to Culp. He is dead serious — rather, "ded serius" — and he speaks passionately about the foibles of language.

He has reduced the alphabet to 25 letters — eliminating "q" —and says other changes will be necessary to reform the world's second most popular language. But changes can't come quickly, he warns, or they won't be accepted.

Changing "city" to "siti" — eliminating the soft "c" and replacing it with "s" — will be accepted, he said, but replacing the hard "c" with a "k" will be tough because that would produce *O Kanada.*

Culp resists the temptation to teach the Canadian "langwaje" to his high school students, but does not hesitate to point out the problems with English.

"It wil spred arownd the wurld," Culp says. "It wil triumf."

Ed note: Now thars a Cannuck thet nos wot hez talken bout! Ether thet er ole George hez ben leten him reed hiz kopi uv Update Magazine. Ile bet thets whut hapend, nunthles hez on mi bandwagun. We gota dew sumthin fore th kids an thar rap talk taks ovr fer sur. -BJ_

The News is Out!

AND WE'RE SPREADING IT!

RMG is adding so many NEW and EXCITING ITEMS to our line that our storerooms are BULGING! We have added more that 25 NEW PAGES and changed over 20 others in our BIG 40+ page catalog. We want to make sure that you do not miss out on anuthing we may be able to help you with in your computing. If you will send us $3.00 you will receive our catalog with a $3 off coupon for your first order. AND, send along 12 business size SASEs and you will receive 12 monthly updates and special mailings. $5 discount coupon in first mailing!

RMG ENTERPRISES
1419 1/2 7TH STREET
OREGON CITY, OREGON 97045
503/655-7484 * NOON-10 TUE-SAT

PLATYPUS Software — QL to PC file transfers

SPREADSHEET [text & valid formulas / some functions do not transfer to LOTUS (e.g., askt & askn)]:

QL-ABACUS to:	PC-ABACUS	base + 1.60 / screen
	LOTUS 1-2-3	base + 4.80 / screen
QL-QUILL to:	PC-ASCII.TXT	base + 2.50 / 25 lines
	PC-QUILL	base + 3.00 / page
	WordPerfect	base + 5.00 / page

The "base" fee is: $5.00 for the 1st file, $4.00 for the 2nd file, $3.00 for the 3rd file, $2.00 for the 4th file, $1.00 for each subsequent file.

A "screen" is defined as: 20 rows x 8 cols.
A half-screen is: 10 rows x 8 cols.
A quarter-screen is: 5 rows x 8 cols.

** A "page" is defined as: 30 lines maximum x 80 columns (or, about 300 words).
** Partial pages will be rated at $0.10 per line in QUILL and $0.15 in WordPerfect.
** A quarter-screen minimum increment for spreadsheets.

PLEASE NOTE: The new file(s) will be supplied on a 3.5" 720K disk.

If you need a 5.25" 360K disk, then please supply a labelled, pre-formatted 360K disk along with any original media. If you send microcartridges, please ensure that they are of high quality AND that you retain backup copies during the interim.

Of course, any original media (disks/microcarts) will be returned.

If you have other needs, please enquire: AL FENG
15 Wake Robin Court
Woodridge, IL 60517
(708) 971-0495

THE QL ISSUE DISK PROGRAM

And how goes it, QL?

The issue disk program is designed to give publicity to the programmers who are working for us and to put the Author into direct contact with the users. Normally, an issue disk title is offered in a couple of issues of the magazine and then rotated out. This introduces the programmer to the users and establishes rapport between them. The programs in the issue disk belong to the authors. Update's only involvement is in their early distribution "at introductory prices".

NOTICE that Bill Cable (Wood and Wind Computing) now has the capability of providing his software on both 5 1/4" and 3 1/2" disks (and microdrive cart). See his ad else- where in this issue. Also NOTICE the ad by AL FENG (Platypus Software). Both of these authors are producing software for the QL and IBM systems.

Other North American software authors will be introduced to the users as they come aboard. The issue disk program's purpose is to encourage the creation of NORTH AMERICAN SOURCES in support of Sinclair Computing. The program is enjoying moderate success. Two years ago Peter Hale (EMSOFT) was about the only recognized source of American Software for the QL. Now he is joined by others. (See EMSOFT Ad in this issue).

While the Brits are jolly good fellows, and they have given us (sold us) many good softwares, Update Magazine takes the position that QL users in North America should not have to swim the pond to get useful software. Our own programmers are available for call to service the users, and are eager to do so.

Didja know that PSION, the producer (emeritus) of our Four great QL Softwares, is now an American Corporation? AND, Bill Cable has a working relationship with PSION to do follow-on improvement programming with the PSION suite? AND, that EMSOFT (Peter Hale) is a Dealer for the new PSION Mobil computers? AND, I'm squirreling away saving to buy one of the new PSION Mobile computers? I dont know what thats got to do with Issue Disks, but it is all relative to the subject of Hobby computing. ENJOY!

SINCLAIR QL ISSUE DISKS (MDV)

The following new issue disks are offered for QL systems of all configurations. The programs are on either 80 track Diskette or MDV cartridge. If ordering for MDV send a formatted MDV cartridge. All prices are $22.00 (post paid) and are guaranteed to satisfy.

---

1. "HARTUNG COLLECTION" DISK

Address and QSO file managers, plus a raft of disk utilities. All in Superbasic. Lots of tricks and hints for the programmers. Requires some knowledge of Superbasic or a yen to learn.

2, "FENG QLUTer" DISK

QLUTer - QL UTilities to un-clutter your disks and tapes. COPY, DELETE, FORMAT, PRINT, VIEW functions. TURBO compiled for SPEED. Multitasks and minimal key press. Also has-- SNAP SHOT - a directory column or condensed print out program. ALSO includes "VEGEMAT" - a CLONE making program.

---

3. "FENG QLUSTer" DISK

QLUSTer - very similiar to QLUTer (above) but has extended functions using some SUPER TOOLKIT II features. You must have the Toolkit (TK2_EXT) for this program. SNAP SHOT and VEGEmat also included.

---

4. CABLE ARCHIVE DISK

ARCHIVE utilities and exensive doc file instruction manuals. Programs are: REFIELD, SCAN, SPLIT, JOIN, DIR, FILER, FREQ. For Beginner to Advanced user. These programs GREATLY EXTEND the capabilities of ARCHIVE and the manuals lead one to complete understanding of Archive and these extensions.
If ordering for MDV send two mdv cartridges- too many programs to fit into one cart.

---

5. "CABLE HELPER2" DISK

A very useful TOOLKIT program for managing both DISKS and MICRODRIVE. HELPER lets you copy from one medium to another, Read or Copy doc files to paper or screen, repeat format mdv carts or floppy. Uses screen windows to see two directories from and to, copies over existing files in disk mdv, copy selected files. Does not conflict with TK2.

Order to UPDATE Magazine, address on cover.
Check or money order accepted.

THE TS-2068 ISSUE DISK PROGRAM

Available until July 31, 1990

While Frank Davis intends to continue the Issue Disk program it is likely to undergo change. The software offered in these disks belong to the authors and new agreements will be required. Some titles may not be available after July.

It has been difficult to pursuade and cajole NEW TS-2068 SOFTWARE from the few programmers who are still active. Most seem to be modifying older cassette software for disk drive operation. Thats kinda like modifying a row-boat to become a cruise ship.

But this issue brings an excellent new BUSINESS Software, by Jim Bretz. My own softwares, Daisy, Smart Text, and Dbasel, were thrown into the Issue disk program to fill a void of new DISK BASED Software. These titles will be withdrawn after July because all are constantly undergoing upgrades, manual addendums, and require personal support via letters and telephone.

My Fortune 800 Software house, "Gulf Micro" will continue to work on these and provide new full-length software as time goes on. I wish that other programmers would join in such efforts. Larry Kenny and John McMichael are doing some good NEW software. AND, the Toronto TSUG has some very active programmers who are producing LKDOS utilities. We need to entice others to join in. Software designed from the ground up for DISK DRIVE SYSTEMS is what we need. Old Cassette software with the SAVE line modified wont fit our needs, and if thats all the users have available they will move on to a computer system that has the software.

A lot of new programming is going into the area of EXTRA MEMORY usage. Larmour Crawford is predominant in this area and I'm trying to assist. Please read the article "The QL ISSUE DISK PROGRAM". Our TS-2068 issue disk program has not had as much success, but lets keep on trying. -BJ_

The Update TS-2068 Disk Library

The following Disk software and utilities are offered. The individual disk listings are given by TYPE of DOS that the disks are prepared for.

The disks are available on 5-1/4" Double Side Double Density Diskettes, either 40 Track (360K) or 80 Track (720K) disk drives. If you have a 5-1/4 inch disk drive these disks will work in your system. Each disk has at least 300K of programs and utilities. The price is $22.00 per disk (postage paid) (add $1.00 to Canada). Checks or Money Order.

DAISY.B6 : A Complete TS-2068 Software System, Data base management, Word Processing, Automatic formatting, Disk Management. Over 20 integrated programs, including a Mail Merge annex. For LKDOS Only. (Oliger DOS users order Smart Text.) An Update Magazine Software.

SMART TEXT : A versatile Data Base and Word Processor software. Includes a Mail Merge annex to create mailing lists for automatic repeat letters. Label printing, Disk Management. By Update Magazine. For Oliger DOS Only.

VIEW CALC UPGRADE : The old PSION View Calc program Expanded and Converted for LKDOS with Spread Sheet Printer. By Bob Mitchell. For LKDOS Only.

1988 UPDATE COLLECTION: The Mail Merge Software plus the disk filled with utilities. For LKDOS or Oliger DOS

1989 UPDATE COLLECTION: Filled with all of the utility programs published during 1989. LKDOS or Oliger DOS

SMART TEXT 64: (Coming) Smart Text software in 64 Colum Screen for use with the Zebra OS-64 Cartridge. For OLIGER DOS ONLY. (Do not order until release is announced)

DBASE-1 : (Coming) A new RECORD software. The data base will be very flexible with applications for Mail Lists, Inventories, or just about any purpose one needs. Sort by any line, and by first or last group within the line. A "Sort Disk Files" feature will allow one to create and sort thousands of continuous records. (Do not order until release is announced.)

1990 UPDATE COLLECTION: All of the Utility programs being published during this current year. (to be released in Oct.)

BUSINESS ACCOUNTS: *New Major Business Software for the TS-2068, By: Jim Bretz*

This software is given as a key-in project and can be ordered for imediate use. See the article in this issue. Available for LKDOS OR SPDOS. 5 1/4" DSDD 80 TRACK 720K FORMAT. Use your own Printer Driver Code or LKDOS version 3 driver. Needs 64 Colum support by either TASWI or Zebra OS-64 cartridge.

NOTE TO PROGRAMMERS: *YOU* can join in with the Issue Disk Program. All you need to do is to write a "Mini-Manual" for your software to be published, and supply the disk.

BACK ISSUES OF UPDATE MAGAZINE are available in YEAR Groups for $18.00 per year (post paid) (add $1.00 to Canada) Each year group has over 200 pages of mostly useful programs and utilities as key-in projects and MANY Tutorials for Basic and MC programming. Available are:

1988 Year Group (All TS-2068 Supportive)

1989 Year Group (TS-2068 and Sinclair QL (LIMITED STOCK REMAINING)

LKDOS SOFT-WARE

MAXCOM 300/1200 baud Terminal/BBS

This full feature modem software lets you effectivly link the modem to the disk, so you can upload or download files larger than 100K.This lets you send NMI saves and large text files as well as Basic or Code files, which other modem programs couldn't send because of memory limitations. Also files can be sent with or without a information header (Spec-term compatible) .

It also has a 64 column display and is the only 2068 modem program that will run at full 1200 baud without missing characters. Other features of the terminal mode are IBM graphic characters, auto-dialer, macro keys, 110K disk buffer for captured text, auto repeat keys.

The BBS is the most elaborate available for the 2068 and allows the remote user full access to the disk drives similar to a CPM BBS. Features of the BBS are - 300/1200 baud, operation with fast reponce even at 1200 baud, Passwords, priority levels, Use time limits, Multiple Message Bases, Uploads and Downloads are not limited by Memory, Dual Window Monitor screen, Interupt driven time out and Hangup protection, Also Keeps a User log of callers Names and Activity on disk so a printer isnt needed and its easily customised and expanded.

MAXCOM can be used with the 2050 modem or with the Z-SIO RS232 interface and a 300/1200 baud modem. (a version for the Aerco RS232 is in the works).

PRICE: $24.95

LARKEN DISK EDITOR

This program lets you examine or change any byte on a LKdos disk. It is useful in repairing damaged or glitched disks, changing the disk Name or head speed etc. You can also examine the track map and disk info headers.

The documentation for the disk editor also includes information on how to access LKdos from Machine Code and information on the structure of the disk catalog and data blocks.

PRICE: $15.00

SEQUENTIAL/RANDOM Access FILES

This utility is a ram resident extension to the LKdos operating system and lets you create, read from, or write to very large data files kept on disk. It uses the commands OPEN#, CLOSE#, PRINT#, INPUT#, and INKEY$# to access the files from BASIC. There is also a high speed search command for finding a text string within the data file. Using the TAB command, you can randomly access directly any record in a file.

The utility is easily used within your basic programs and has a lot of uses, from storing large data files, creating large spell checker dictionarys or even converting you basic programs to an ascii text file .

This utility lets you take the limits off of the data capacity of the 2068 .Files on disk can be huge !

A simple data base program is included on the disk to demonstrate the utility.

PRICE: $ 15.00

Coming Soon from Larken Electronics

** The Larken Desk Top Publisher ** Produce high quality printing (like this page) with an epson compatible printer and LKdos. The entire Hi-res graphic page is kept on the disk and scrolling though the page is as easy as clicking the joystick or mouse on the icon type menu. Text and Graphic editing with multiple sized fonts will also be featured.

** Spell Checker for Tasword and M-Script ** This utility program will let you have your wordprocessor text files be checked by a 350K dictionary.

All Prices $US - Specify type of drive when ordering.

LARKEN ELECTRONICS RR#2 NAVAN, ONTARIO CANADA, K4B-1H9

(613)-835-2680